TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG
KHOA CÔNG NGHỆ THÔNG TIN

BÁO CÁO CUỐI KỲ

MÔN DEEP LEARNING

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

Người thực hiện: **TRẦN ANH KIỆT – 52300124**

TRẦN THẢO MY – 52300129

Nhóm môn học**:** **02**

Năm học**:** **2025-2026**

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

BÁO CÁO CUỐI KỲ

MÔN DEEP LEARNING

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

Người thực hiện: **TRẦN ANH KIỆT – 52300124**

TRẦN THẢO MY – 52300129

Nhóm môn học**:** **02**

Năm học**:** **2025-2026**

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2025

LỜI CẢM ƠN

Lời đầu tiên, chúng em muốn dành những lời tri ân chân thành đến Trường Đại học Tôn Đức Thắng vì đã đưa môn học "Nhập môn học sâu" vào chương trình giảng dạy. Đặc biệt, chúng em xin gửi lời cảm ơn sâu sắc đến thầy Lê Anh Cường - người đã không chỉ là giảng viên mà còn là người hướng dẫn tận tâm, truyền đạt những kiến thức quý báu cho chúng em suốt thời gian học tập vừa qua.

Trong những buổi học của Thầy, chúng em đã được trải nghiệm sự chuyên nghiệp và tận tâm của người hướng dẫn. Nhờ sự giảng dạy và hướng dẫn kỹ lưỡng của thầy, mà chúng em đã nắm vững những khái niệm, thuật toán, cũng như cách tiếp cận bài toán một cách sáng tạo. Thầy không chỉ truyền đạt kiến thức mà còn khích lệ chúng em phát triển tư duy logic và kỹ năng giải quyết vấn đề.

Môn học "Nhập môn học sâu" không chỉ thú vị mà còn rất bổ ích và thực tế. Chúng em tin rằng những kiến thức chúng em đang học sẽ là hành trang quan trọng, giúp chúng em tự tin bước vào lĩnh vực Công nghệ thông tin sau này. Dù chúng em đã nỗ lực hết mình, nhưng do vẫn còn những hạn chế và khó khăn, bài tiểu luận của chúng em không tránh khỏi những thiếu sót. Chúng em kính mong thầy xem xét và góp ý để bài tiểu luận trở nên hoàn thiện hơn.

Chúng em xin chân thành cảm ơn!”

TP. Hồ Chí Minh, ngày 7 tháng 11 năm 2025

Tác giả

Trần Anh Kiệt

Trần Thảo My

CÔNG TRÌNH ĐƯỢC HOÀN THÀNH

TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG

Chúng tôi xin cam đoan đây là công trình nghiên cứu của riêng chúng tôi và được sự hướng dẫn khoa học của Thầy 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 Dự 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 Dự 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 7 tháng 11 năm 2025

Tác giả

Trần Anh Kiệt

Trần Thảo My

MỤC LỤC

Phần 1: Giới thiệu đề tài.

1.1 Bối cảnh.

- Trong vài năm gần đây, các mô hình ngôn ngữ lớn (Large Language Models - LLMs) như GPT, LLaMA, Falcon, Mistral, Vicuna… đã đạt được những tiến bộ vượt bậc trong các tác vụ xử lý ngôn ngữ tự nhiên (NLP) như:

- Dịch máy.
- Hỏi đáp tự động.
- Tóm tắt văn bản tự động.
- Viết nội dung tự động.

- Các mô hình sinh chuỗi token (sequence generation models) đã tạo nên một cuộc cách mạng trong lĩnh vực Trí tuệ Nhân tạo. Khả năng sinh ra văn bản, mã nguồn, hay các chuỗi dữ liệu có cấu trúc một cách mạch lạc và phù hợp ngữ cảnh đã mở ra vô số ứng dụng thực tiễn, từ dịch máy, tóm tắt văn bản, đến các trợ lý ảo thông minh. Cốt lõi của các mô hình này là kiến trúc Transformer, và sức mạnh của chúng được khuếch đại thông qua việc tiền huấn luyện (pre-training) trên các tập dữ liệu khổng lồ.

1.2 Bài toán: Tóm tắt văn bản tiếng Việt.

- Báo cáo này tập trung vào việc xây dựng và tinh chỉnh một mô hình ngôn ngữ lớn để tự động tóm tắt văn bản tiếng Việt.

- Tóm tắt văn bản là bài toán nhận đầu vào là một đoạn hoặc bài viết dài, và sinh ra một đoạn văn ngắn hơn thể hiện các ý chính, đồng thời đảm bảo độ chính xác ngữ nghĩa và tính mạch lạc trong cách diễn đạt.

- Tiếng Việt, với đặc điểm ngữ pháp linh hoạt, dấu thanh và phong cách đa dạng, đặt ra thách thức lớn cho các mô hình vốn chủ yếu được tiền huấn luyện bằng tiếng Anh.

- Mục tiêu của đề tài là phát triển một mô hình có khả năng:

- Hiểu và rút gọn nội dung của văn bản tiếng Việt.
- Sinh ra bản tóm tắt ngắn gọn, chính xác và tự nhiên, giống như cách con người tóm tắt.

1.3 Lựa chọn Mô hình và Lộ trình Kỹ thuật.

- **Mô hình Nền (Base Model)**: Qwen/Qwen3-0.6B, một mô hình ngôn ngữ mới cuả Alibaba (600 triệu tham số), được xây dựng trên kiến trúc Transformer Decoder-Only. Mô hình này hỗ trợ nhiều ngôn ngữ, bao gồm cả tiếng Việt, có hiệu năng tốt trong các tác vụ sinh văn bản.
- **Kỹ thuật tối ưu hóa:**
  - SFT (Supervised Fine-Tuning): Huấn luyện mô hình trên một tập dữ liệu chứa các cặp (văn bản → bản tóm tắt) để dạy mô hình kỹ năng rút gọn nội dung.
  - DPO (Direct Preference Optimization):

- **Cấu trúc Báo cáo:** Báo cáo sẽ trình bày một cách hệ thống quy trình xây dựng mô hình, từ việc lựa chọn và xử lý dữ liệu, triển khai các kỹ thuật SFT và DPO, cho đến việc đánh giá toàn diện hiệu suất của ba phiên bản: mô hình pre-trained gốc, mô hình sau khi SFT, và mô hình sau khi DPO.

Phần 2: Mô hình pre-train và thực hiện tinh chỉnh có giám sát (Supervised Fine-Tuning - SFT).

2.1 Lựa chọn mô hình pre-trained.

- Kiến trúc: Qwen3-0.6B là một mô hình thuộc họ Qwen2, được xây dựng trên kiến trúc Transformer Decoder-Only, sử dụng cơ chế Self-Attention để mô hình hóa ngữ cảnh.
- Quy mô: khoảng 600 triệu tham số, tối ưu cho tốc độ huấn luyện và triển khai trên GPU phổ thông.
- Ưu điểm:
  - Hỗ trợ đa ngôn ngữ, bao gồm tiếng Việt.
  - Mô hình nhỏ nhưng hiệu năng cạnh tranh với các model lớn hơn.
  - Dễ dàng kết hợp với các kỹ thuật PEFT như LoRA, Quantization để giảm chi phí huấn luyện.

2.2 Kỹ thuật Tinh chỉnh Hiệu suất cao (PEFT): LoRA và lượng tử 4 bit.

- Trước hết, việc lượng tử hóa 4-bit (với tham số load_in_4bit=True) cho phép nén các trọng số từ định dạng 32-bit hoặc 16-bit xuống 4-bit, giúp giảm dung lượng bộ nhớ cần thiết khi tải mô hình. Điều này cho phép mô hình Qwen3-0.6B được huấn luyện ngay cả trên các GPU phổ thông có dung lượng VRAM hạn chế.

- Trong quá trình thực nghiệm, mô hình được tinh chỉnh theo hướng Supervised Fine-Tuning (SFT) kết hợp với PEFT (Parameter-Efficient Fine-Tuning), cụ thể là phương pháp LoRA (Low-Rank Adaptation).

- Mô hình nền sử dụng là Qwen/Qwen3-0.6B, thuộc nhóm mô hình ngôn ngữ nhân tạo kích thước nhỏ (khoảng 600 triệu tham số), được tối ưu cho các tác vụ sinh văn bản (Causal Language Modeling).

- Trong quá trình tinh chỉnh:

- Mô hình gốc được tải bằng AutoModelForCausalLM.from_pretrained(...) ở chế độ precision mặc định (FP16/FP32).
- Phương pháp LoRA được áp dụng nhằm giảm số lượng tham số cần huấn luyện. Thay vì cập nhật toàn bộ trọng số của mô hình, LoRA chỉ thêm một số ma trận trọng số có hạng thấp (low-rank) vào các lớp cần điều chỉnh, từ đó giảm đáng kể chi phí bộ nhớ và thời gian huấn luyện.
- Việc tinh chỉnh được thực hiện thông qua Trainer và TrainingArguments của thư viện Hugging Face Transformers, cùng với DataCollatorForLanguageModeling để tạo batch dữ liệu phù hợp cho mô hình sinh chuỗi.
- Sau khi huấn luyện, các trọng số LoRA được lưu riêng biệt. Trong giai đoạn đánh giá (inference), mô hình gốc được nạp lại và bọc qua PeftModel.from_pretrained(...) để gắn thêm các tham số LoRA đã huấn luyện, giúp mô hình thực hiện tốt hơn trên tác vụ tóm tắt văn bản.

Cấu trúc kỹ thuật này mang lại ưu điểm lớn: giúp mô hình đạt hiệu quả huấn luyện tương đương fine-tuning đầy đủ, trong khi tiết kiệm tài nguyên tính toán, bộ nhớ GPU và thời gian huấn luyện.

2.3 Tập dữ liệu và Quy trình Tiền xử lý.

2.3.1 Giới thiệu về Tập dữ liệu OpenHust/vietnamese-summarization.

- Đối với bài toán tóm tắt tiếng Việt, chúng em sử dụng tập dữ liệu OpenHust/vietnamese-summarization, được công bố trên Hugging Face Hub.

- Mỗi mẫu dữ liệu bao gồm hai trường: "text" chứa đoạn văn hoặc bài viết gốc, và "summary" là bản tóm tắt tương ứng. Trung bình, mỗi đoạn văn có độ dài từ 150–400 từ, trong khi phần tóm tắt chỉ khoảng 30–60 từ, thể hiện tỷ lệ nén thông tin vào khoảng 1:6 đến 1:8. Dữ liệu bao phủ nhiều chủ đề khác nhau như đời sống, giáo dục, chính trị, công nghệ, và văn hóa, giúp mô hình học được cách rút trích nội dung trong nhiều ngữ cảnh khác nhau.

- Trong khuôn khổ dự án, để phù hợp với tài nguyên tính toán, chỉ khoảng 7.000 mẫu (tương đương 3% dữ liệu gốc) được sử dụng để huấn luyện. Dữ liệu được chia ngẫu nhiên thành ba phần: 80% cho huấn luyện, 10% cho kiểm thử và 10% cho đánh giá. Việc chọn mẫu được thực hiện sao cho đảm bảo độ đa dạng chủ đề và độ dài văn bản.

2.3.2 Quy trình Tiền xử lý.

- Định dạng mẫu (instruction-like format: Mỗi cặp (Document, Summary) được chuyển thành một chuỗi đầu vào duy nhất theo template (được tạo trong hàm preprocess_function). Mục đích là đưa dữ liệu về dạng "instruction + input + output" giúp SFT dễ học cách sinh ra summary.

```
full_text = f"### Input:\n{input_text}\n\n### Summary:\n{summary_text}{tokenizer.eos_token}"
```

- Tokenization: Những chuỗi full_text được tokenize bằng tokenizer của Qwen (tokenizer được truyền vào hàm):

```
tokenized = tokenizer(
    texts,
    max_length=1024,   # giới hạn chiều dài token
    truncation=True,   # cắt nếu vượt quá max_length
    padding=False,     # không padding ở bước preprocessing
```

```
    return_tensors=None
)
```

max_length=1024 (giới hạn chuỗi token đầu vào).

truncation=True để cắt các văn bản quá dài.

- Ở giai đoạn map preprocessing notebook không áp padding, padding được xử lý sau (hoặc bằng data-collator của Trainer).

- Tạo nhãn (labels): Sau khi token hóa, notebook gán labels bằng một bản sao của input_ids.

```
tokenized["labels"] = tokenized["input_ids"].copy()
```

- Điều này phù hợp với việc huấn luyện dạng causal LM trên chuỗi "prompt + expected output" (mô hình học sinh cả prompt lẫn phần đáp ứng; khi training sẽ sử dụng mask/shift để tính loss chỉ trên phần cần dự đoán).

- Áp dụng map lên dataset: Hàm preprocessing được áp dụng theo batch lên dataset

```
tokenized_ds = ds.map(
    preprocess_function,
    batched=True,
    batch_size=1000,
    remove_columns=ds.column_names,
    num_proc=1
```

```
    )
```

- remove_columns=ds.column_names để chỉ giữ các trường tokenize (input_ids, attention_mask, labels, ...).
- batched=True với batch_size=1000 để tăng tốc.

- Thống kê tokenization: tổng token, trung bình, max/min, và tập độ dài sequence (unique lengths) để kiểm tra việc cắt/padding.

```
    input_lengths = [len(x['input_ids']) for x in tokenized_ds]

    # in avg, max, min, unique lengths
```

- Lưu ý về padding / dynamic padding.

- Trong preprocessing, đặt padding=False (không padding). Trong quá trình huấn luyện, padding thường được xử lý bởi DataCollatorForLanguageModeling hoặc Trainer
- Như vậy, padding sẽ xảy ra ở runtime batch-collation (để giảm waste do padding trước khi gửi vào GPU).

2.4 Quá trình tinh chỉnh.

2.4.1 Chuẩn bị mô hình và kỹ thuật LoRA.

- Đầu tiên, mô hình Qwen được nạp ở dạng lượng tử hóa 4-bit nhằm giảm dung lượng bộ nhớ GPU cần thiết. Sau đó, kỹ thuật Low-Rank Adaptation (LoRA) được áp dụng để chỉ tinh chỉnh một phần nhỏ tham số của mô hình, giúp tiết kiệm tài nguyên trong khi vẫn đạt được khả năng thích ứng cao. Các lớp được chọn để áp dụng LoRA gồm q_proj, k_proj, v_proj, và o_proj trong khối Attention – đây là những trọng số có ảnh hưởng mạnh nhất đến khả năng biểu diễn của mô hình.

2.4.2 Chuẩn bị dữ liệu và tiền xử lý.

- Dữ liệu được lấy từ bộ OpenHust/vietnamese-summarization, gồm các cặp văn bản gốc (“Document”) và bản tóm tắt (“Summary”).
- Mỗi mẫu được định dạng lại thành dạng hướng dẫn (instruction format):

```
### Input:
[Nội dung văn bản]
### Summary:
[Tóm tắt ngắn gọn]
```

- Cấu trúc này giúp mô hình hiểu rõ ranh giới giữa phần đầu vào và phần cần sinh. Văn bản sau đó được token hóa với max_length=1024 token, cắt ngắn nếu vượt quá giới hạn, và gán labels = input_ids để phục vụ huấn luyện dạng causal language modeling.

- Trong phiên bản cải tiến, các token thuộc phần "Input" được gán nhãn -100 trong labels để loại khỏi hàm mất mát (loss). Nhờ đó, mô hình chỉ học sinh ra phần "Summary" thay vì cả prompt, giúp chất lượng sinh tóm tắt tự nhiên và chính xác hơn.

2.4.3 Cấu hình huấn luyện.

- Việc huấn luyện sử dụng Hugging Face Trainer với các tham số chính:

- learning_rate = 1e-4
- per_device_train_batch_size = 2, gradient_accumulation_steps = 4 (batch hiệu dụng = 8)
- num_train_epochs = 2
- fp16 = True để tăng tốc và giảm bộ nhớ
- gradient_checkpointing = True để tiết kiệm VRAM
- save_strategy = "epoch" nhằm lưu checkpoint sau mỗi epoch.

2.4.4 Kết quả và mô hình đầu ra.

- Sau khi tinh chỉnh, các tham số LoRA được lưu riêng thành adapter có kích thước nhỏ (~ vài chục MB).
- Khi gắn adapter này lên mô hình gốc Qwen, mô hình có thể sinh ra các bản tóm tắt tiếng Việt tự nhiên, mạch lạc hơn rõ rệt so với mô hình ban đầu.
- Quá trình này cũng chứng minh hiệu quả của việc sử dụng QLoRA + Instruction-format fine-tuning trong điều kiện tài nguyên hạn chế.

Phần 3: Áp dụng phương pháp Reinforcement Learning Direct Preference Optimization (DPO).

3.1 Giới thiệu về phương pháp DPO.

- Sau khi hoàn thành giai đoạn tinh chỉnh có giám sát (Supervised Fine-Tuning – SFT), mô hình đã học được cách sinh ra bản tóm tắt tương đối chính xác về mặt ngữ nghĩa. Tuy nhiên, bản tóm tắt sinh ra có thể vẫn còn cứng nhắc, chưa hoàn toàn phù hợp với tiêu chí "tự nhiên", "mạch lạc" hoặc "được người dùng ưa thích".
Để giải quyết điều này, nhóm đã áp dụng Direct Preference Optimization (DPO) – một phương pháp học tăng cường (Reinforcement Learning from Human Feedback – RLHF) đơn giản hóa, giúp mô hình học cách ưu tiên những đầu ra được đánh giá là tốt hơn mà không cần mô hình phần thưởng (reward model) như các phương pháp truyền thống (ví dụ: PPO – Proximal Policy Optimization).

3.2 Nguyên lý của DPO.

- Phương pháp DPO (Rafailov et al., 2023) trực tiếp tối ưu xác suất sinh ra phản hồi được ưa thích (chosen) so với phản hồi kém (rejected), thông qua hàm mục tiêu:

$$\mathcal{L}_{DPO} = -\log \sigma \left(\beta \cdot \left(\log \pi_\theta(y^+|x) - \log \pi_\theta(y^-|x) \right) \right)$$

- Trong đó:
  - y+: bản tóm tắt được yêu thích hơn.
  - y-: bản tóm tắt bị từ chối (rejected).
  - π_0: phân phối đầu ra của mô hình.
  - β: hệ số điều chỉnh độ nhạy.
- Khác với PPO, DPO không cần ước lượng phần thưởng (reward model) hay đạo hàm giá trị (advantage), mà trực tiếp huấn luyện mô hình chính để phân biệt hai phản hồi trên cùng một prompt.

3.3 Quy trình huấn luyện.

3.3.1 Tạo dataset DPO từ dataset sẵn có.

```
ds=load_dataset("OpenHust/vietnamese-summarization", split="train[:300]")
```

- **Ý tưởng chính:** Lấy một phần dữ liệu tóm tắt tiếng Việt (ở đây là 300 mẫu đầu tiên) để làm dữ liệu huấn luyện DPO.
- **Tạo cặp** chosen **và** rejected**:**

```
prompt = f"Tóm tắt văn bản sau: {sample['Document']}"

chosen = sample['Summary']

rejected = create_bad_summary(sample['Document'])
```

  - chosen: summary chất lượng từ dataset.
  - rejected: summary "kém chất lượng", tạo bằng cách cắt ngắn, chung chung, hoặc vô nghĩa. Đây là cốt lõi của DPO, vì DPO học để mô hình ưu tiên chosen hơn rejected.

- Ý nghĩa: Tạo dữ liệu huấn luyện DPO, nơi mô hình học từ sự ưa thích giữa output tốt và output xấu thay vì chỉ học "input → output".

3.3.2 Lưu dataset thành CSV.

```
df.to_csv("dpo_dataset.csv", index=False, encoding='utf-8')
```

- Giúp lưu dataset DPO để có thể dùng trực tiếp cho huấn luyện.
- CSV gồm các cột:

  - prompt: câu hỏi/gợi ý cho mô hình.
  - chosen: kết quả đúng, mong muốn.
  - rejected: kết quả kém, không mong muốn.

3.3.3 Chuẩn bị mô hình và tokenizer.

```
tokenizer = AutoTokenizer.from_pretrained(BASE_MODEL, use_fast=True)

model = AutoModelForCausalLM.from_pretrained(BASE_MODEL)
```

```
model = PeftModel.from_pretrained(model, SFT_PEFT_PATH)
```

- **Ý tưởng chính:**
  - Sử dụng **mô hình Qwen3-0.6B** làm base model.
  - Áp dụng **SFT adapter** (mô hình đã được fine-tune trước đó) để giảm số lượng tham số cần huấn luyện.
  - Tokenizer chuẩn bị để mã hóa văn bản thành token, phục vụ huấn luyện và sinh văn bản.

- **PEFT (LoRA/Adapter):**
  - Chỉ huấn luyện một phần nhỏ tham số (LoRA/adapter), giúp giảm VRAM và thời gian huấn luyện.
  - Phần còn lại của mô hình frozen.

3.3.4 Huấn luyện với DPO.

```
training_args = DPOConfig(...)

trainer = DPOTrainer(

   model=model,

   args=training_args,

   train_dataset=dataset,

)

trainer.train()
```

- **Ý tưởng chính:**
  - DPO (Direct Preference Optimization): học từ cặp (chosen, rejected) để mô hình ưu tiên output tốt hơn output xấu, thay vì học trực tiếp target.
  - DPO được thiết kế để fine-tune mô hình ngôn ngữ theo sở thích con người.

- Các tham số quan trọng:

- max_prompt_length, max_completion_length: giới hạn độ dài input/output.
- beta: cân bằng giữa reward của chosen và rejected.
- loss_type="sigmoid": hàm loss DPO.

- **Ý nghĩa:** Mô hình sẽ học cách tóm tắt văn bản theo đúng sở thích/quality mà không cần label trực tiếp về giá trị số của summary.

3.3.5 Kiểm tra mô hình sau huấn luyện.

```
inputs = tokenizer(prompt, return_tensors="pt").to(model.device)
outputs = model.generate(**inputs, max_new_tokens=100)
```

- Kiểm tra mô hình sinh summary từ một đoạn văn bản mới.
- Phần output tách ra từ "Tóm tắt:" để lấy summary thực sự.
- **Ý nghĩa:** Xác minh rằng mô hình đã học được cách tạo summary chất lượng hơn sau khi DPO fine-tune.

Phần 4: Đánh giá và so sánh các mô hình.

Ở phần này ta sẽ trình bày chi tiết về quá trình và kết quả đánh giá hiệu năng của các mô hình đã xây dựng. Mục tiêu là so sánh một cách khách quan khả năng tóm tắt văn bản của ba phiên bản mô hình: Mô hình Pre-trained Gốc, Mô hình sau Supervised Fine-tuning (SFT), và Mô hình sau tinh chỉnh bằng Reinforcement Learning (DPO).

4.1 Phương pháp đánh giá.

Để đánh giá để có cái nhìn toàn diện về chất lượng của các bản tóm tắt, ta sử dụng phương pháp đánh giá tự động. Sử dụng ROUGE (Recall-Oriented Understudy for Gisting Evaluation): Là bộ độ đo tiêu chuẩn để đo lường sự trùng lặp về n-gram. ROUGE gồm ba chỉ số chính

- ROUGE-1: Tỷ lệ trùng lặp từ đơn (unigram).
- ROUGE-2: Tỷ lệ trùng lặp cặp từ (bigram).
- ROUGE-L: Dựa trên chuỗi con chung dài nhất (LCS), đo lường sự tương đồng về cấu trúc câu.

4.2 Kết quả đánh giá.

Mô hình	ROUGE-1	ROUGE-2	ROUGE-L
Pre-trained gốc	0.5104	0.1899	0.2981
Sau SFT	0.5301	0.2294	0.3184
Sau DPO	0.5387	0.2304	0.3169

Phân tích kết quả:

- MQwen SFT (LoRA) và Qwen DPO (RL) đều vượt Qwen gốc rõ rệt, đặc biệt ở ROUGE-2 → khả năng nắm bắt ngữ cảnh và nối kết câu được cải thiện.
- DPO (RL) nhỉnh nhẹ hơn SFT một chút về ROUGE-1 và ROUGE-2, cho thấy học tăng cường giúp mô hình tạo tóm tắt tự nhiên và khớp nội dung hơn.
- Kết luận: SFT và DPO đều cải thiện chất lượng tóm tắt, trong đó DPO đạt kết quả tốt nhất tổng thể.

TÀI LIỆU THAM KHẢO

[1] Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., ... & Zettlemoyer, L. (2020). BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*

[2] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems*