**TÀI LIỆU HƯỚNG DẪN HUẤN LUYỆN MÔ HÌNH CHATBOT SỬ DỤNG DIALOGPT**

**I. Giới thiệu**

Mục tiêu của tài liệu này là hướng dẫn chi tiết quy trình huấn luyện mô hình **DialoGPT**, một biến thể của **GPT-2** được tinh chỉnh đặc biệt cho hội thoại tự nhiên.  
DialoGPT được phát triển bởi Microsoft Research, dựa trên kiến trúc **Transformer Decoder**, được huấn luyện trên hàng triệu đoạn hội thoại Reddit.

**II. Kiến trúc mô hình DialoGPT**

DialoGPT kế thừa kiến trúc từ **GPT-2**:

* Sử dụng **Transformer Decoder-only** gồm nhiều lớp attention và feed-forward.
* Cơ chế **causal self-attention** giúp mô hình hiểu ngữ cảnh theo hướng thời gian (trái → phải).
* Được tinh chỉnh (fine-tuned) trên hơn 147 triệu lượt hội thoại Reddit.

So sánh nhanh:

| **Mô hình** | **Loại Transformer** | **Sử dụng** | **Mục tiêu** |
| --- | --- | --- | --- |
| BERT | Encoder | Hiểu ngữ nghĩa | Classification, NER |
| GPT-2 / DialoGPT | Decoder | Sinh hội thoại, text | Chatbot, text generation |

**III. Yêu cầu môi trường và thư viện**

**1️. Môi trường**

* **Python:** >= 3.8
* **RAM:** >= 8 GB (tốt nhất 16 GB trở lên)
* **GPU:** Khuyến nghị (CUDA hoặc Colab GPU)

**2️. Thư viện cần cài**

***!pip install transformers datasets accelerate pandas torch***

Các thư viện chính:

* **transformers:** để gọi mô hình và tokenizer DialoGPT
* **datasets:** để quản lý và xử lý dữ liệu hội thoại
* **torch:** để huấn luyện mô hình (PyTorch backend)
* **pandas:** đọc file CSV chứa dữ liệu hội thoại
* **accelerate:** hỗ trợ tối ưu huấn luyện trên GPU

**IV. Chuẩn bị dữ liệu huấn luyện**

Dữ liệu thường nằm trong file .csv hoặc .json, gồm các cột ví dụ:

**Các bước xử lý:**

1. Loại bỏ dòng trống, ký tự đặc biệt, emoji nếu không cần.
2. Ghép các câu hỏi – trả lời thành **cặp hội thoại liên tiếp**.
3. Ghi vào danh sách để mô hình học theo chuỗi liên tục (prompt → response).

conversations = []

for i in range(len(df)-1):

if df.iloc[i]["type"] == "user" and df.iloc[i+1]["type"] == "bot":

prompt = df.iloc[i]["message"]

response = df.iloc[i+1]["message"]

conversations.append(f"User: {prompt}\nBot: {response}")

**V. Tokenization và Dataset**

Mô hình sử dụng **tokenizer** tương ứng với DialoGPT:

from transformers import AutoTokenizer

tokenizer = AutoTokenizer.from\_pretrained("microsoft/DialoGPT-medium")

Chuẩn bị dữ liệu cho mô hình:

from datasets import Dataset

train\_dataset = Dataset.from\_dict({

"text": conversations

})

def tokenize\_fn(example):

return tokenizer(example["text"], truncation=True, padding="max\_length", max\_length=512)

tokenized\_dataset = train\_dataset.map(tokenize\_fn, batched=True)

**VI. Huấn luyện mô hình (Fine-tuning)**

Tải mô hình:

from transformers import AutoModelForCausalLM

model = AutoModelForCausalLM.from\_pretrained("microsoft/DialoGPT-medium")

Thiết lập tham số huấn luyện:

from transformers import Trainer, TrainingArguments

training\_args = TrainingArguments(

output\_dir="./chatbot\_model",

evaluation\_strategy="epoch",

learning\_rate=5e-5,

per\_device\_train\_batch\_size=2,

num\_train\_epochs=3,

weight\_decay=0.01,

save\_total\_limit=2,

)

Khởi tạo Trainer:

trainer = Trainer(

model=model,

args=training\_args,

train\_dataset=tokenized\_dataset

)

trainer.train()

**VII. Đánh giá và kiểm thử mô hình**

Sau khi huấn luyện xong, bạn có thể sinh phản hồi mẫu:

from transformers import pipeline

chatbot = pipeline("text-generation", model="./chatbot\_model", tokenizer=tokenizer)

prompt = "Xin chào, bạn có thể giúp tôi tìm vali du lịch không?"

response = chatbot(prompt, max\_length=100, pad\_token\_id=tokenizer.eos\_token\_id)

print(response[0]['generated\_text'])

Hoặc kiểm thử với nhiều prompt:

for question in ["Chào bạn", "Tôi muốn mua balo học sinh", "Cảm ơn nhé"]:

print( question)

output = chatbot(question, max\_length=80)

print(, output[0]['generated\_text'])

**VIII. Lưu và tải lại mô hình**

Lưu mô hình đã huấn luyện:

model.save\_pretrained("chatbot\_model")

tokenizer.save\_pretrained("chatbot\_model")

Tải lại khi cần:

from transformers import AutoModelForCausalLM, AutoTokenizer

model = AutoModelForCausalLM.from\_pretrained("chatbot\_model")

tokenizer = AutoTokenizer.from\_pretrained("chatbot\_model")

**IX. Hướng dẫn tinh chỉnh lại (Retrain)**

Khi có thêm dữ liệu hội thoại mới:

1. Ghép dữ liệu cũ và mới → new\_dataset.csv
2. Tokenize lại toàn bộ dữ liệu
3. Gọi lại trainer.train() từ checkpoint cũ:

trainer.train(resume\_from\_checkpoint=True)

1. Lưu lại phiên bản mới chatbot\_model\_v2

**X. Kết luận và hướng phát triển**

* **DialoGPT** là mô hình mạnh mẽ để tạo chatbot có khả năng phản hồi tự nhiên.
* Dễ dàng tinh chỉnh với dữ liệu hội thoại doanh nghiệp, giúp mô hình học phong cách giao tiếp riêng.
* Hạn chế: đôi khi phản hồi lan man, thiếu kiểm soát.
* Có thể cải tiến bằng:
  + Kết hợp bộ lọc kiểm duyệt nội dung.
  + Sử dụng mô hình **Llama 2 Chat** hoặc **Mistral 7B** để nâng độ chính xác.
  + Tích hợp với cơ sở dữ liệu tri thức hoặc RAG (Retrieval-Augmented Generation).