Tu Pham - Digital Fortress

Large Language Model (LLM) Fine-tuning example

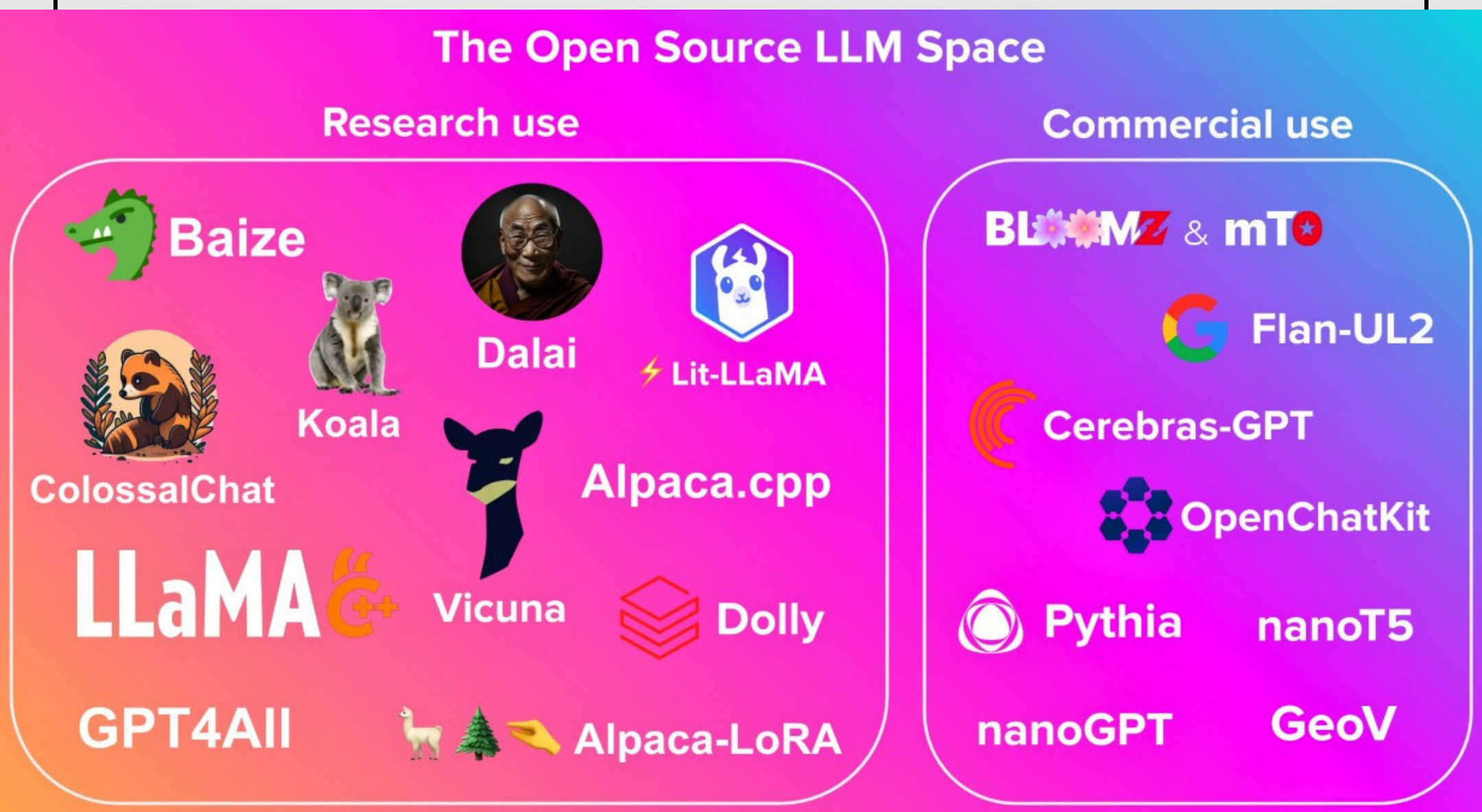
Vol.01

Mô hình ngôn ngữ lớn (LLM) là gì?

Mô hình ngôn ngữ lớn (LLM) là một loại chương trình trí tuệ nhân tạo (AI) có thể nhận dạng và tạo văn bản, cùng với các tác vụ khác. LLM được đào tạo trên các tập dữ liệu khổng lồ - do đó có tên là "lớn". LLM được xây dựng dựa trên học máy: cụ thể là một loại mạng thần kinh được gọi là Transformer Model.

LLM được sử dụng để làm gì?

- **Phân tích tình cảm**
- **Dịch vụ khách hàng**
- **Chatbot**
- **Tóm tắt văn bản**
- **Dịch văn bản**

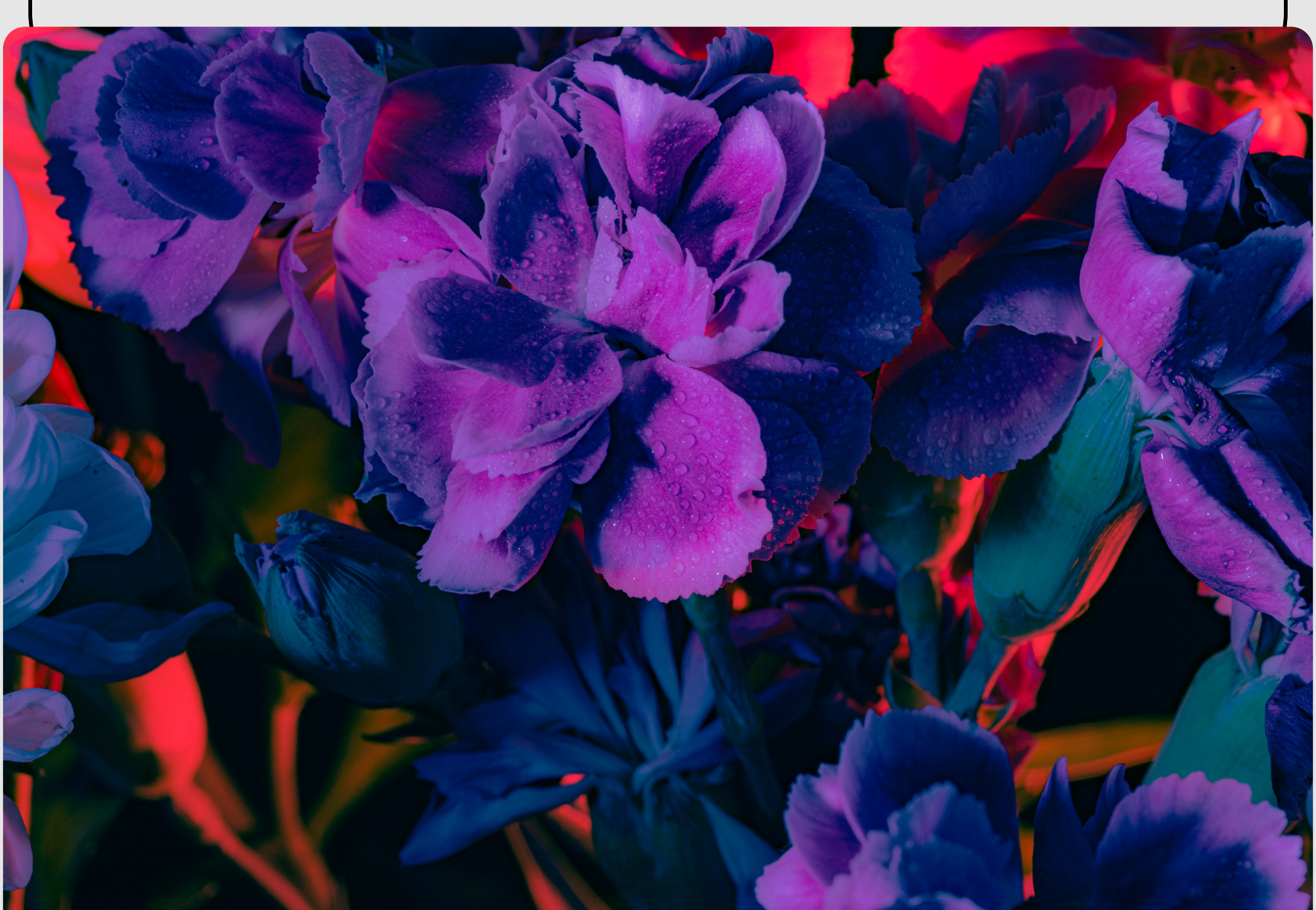


Dưới đây là một số mô hình LLM có thể tham khảo:

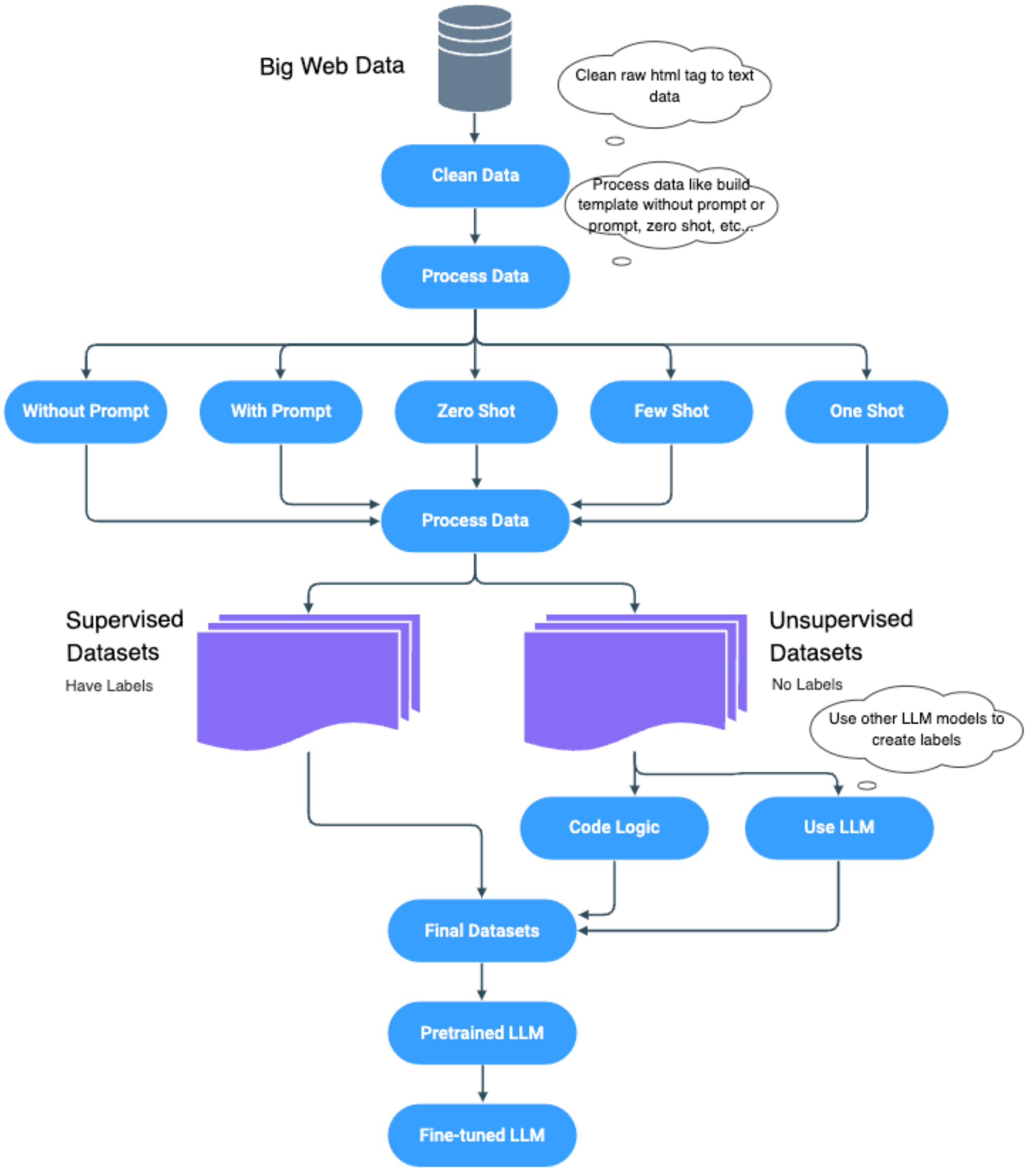
- LLama 2
- Mistral 7B
- Vicuna
- Roberta
- T5
- Flan T5
- Etc..

Workshop hôm nay có gì?

- Các kỹ thuật trong tinh chỉnh (fine-tuning) mô hình LLM bao gồm Zero Shot, Few Shot, etc.
- Giới thiệu về QLora, Pert và Quantize để tinh chỉnh mô hình LLM lớn, rút ngắn thời gian đào tạo.
- Siêu tham số (Hyperparameter), Training Frameworks.
- Notebooks example: Hướng dẫn tinh chỉnh (fine-tuning) trên mô hình Google Flan T5 small.
- Build chat bot đơn giản sau khi tinh chỉnh với Streamlit và LangChain.
- Demo chat bot.



Training Diagram



Kiểm Tra & Đánh Giá Mô Hình

Datasets:

- **GLUE**
- **SuperGLUE**
- **SQuAD**

Example Tasks:

- **MMLU: Massive Multitask Language Understanding.**
- **TriviaQA: TriviaqQA QA.**
- **BoolQ: QA yes/no**

Benchmark (Higher is better)	MPT (7B)	Falcon (7B)	Llama-2 (7B)	Llama-2 (13B)	MPT (30B)	Falcon (40B)	Llama-1 (65B)	Llama-2 (70B)
MMLU	26.8	26.2	45.3	54.8	46.9	55.4	63.4	68.9
TriviaQA	59.6	56.8	68.9	77.2	71.3	78.6	84.5	85.0
Natural Questions	17.8	18.1	22.7	28.0	23.0	29.5	31.0	33.0
GSM8K	6.8	6.8	14.6	28.7	15.2	19.6	50.9	56.8
HumanEval	18.3	N/A	12.8	18.3	25.0	N/A	23.7	29.9
AGIEval (English tasks only)	23.5	21.2	29.3	39.1	33.8	37.0	47.6	54.2
BoolQ	75.0	67.5	77.4	81.7	79.0	83.1	85.3	85.0

Hyperparameter Optimization

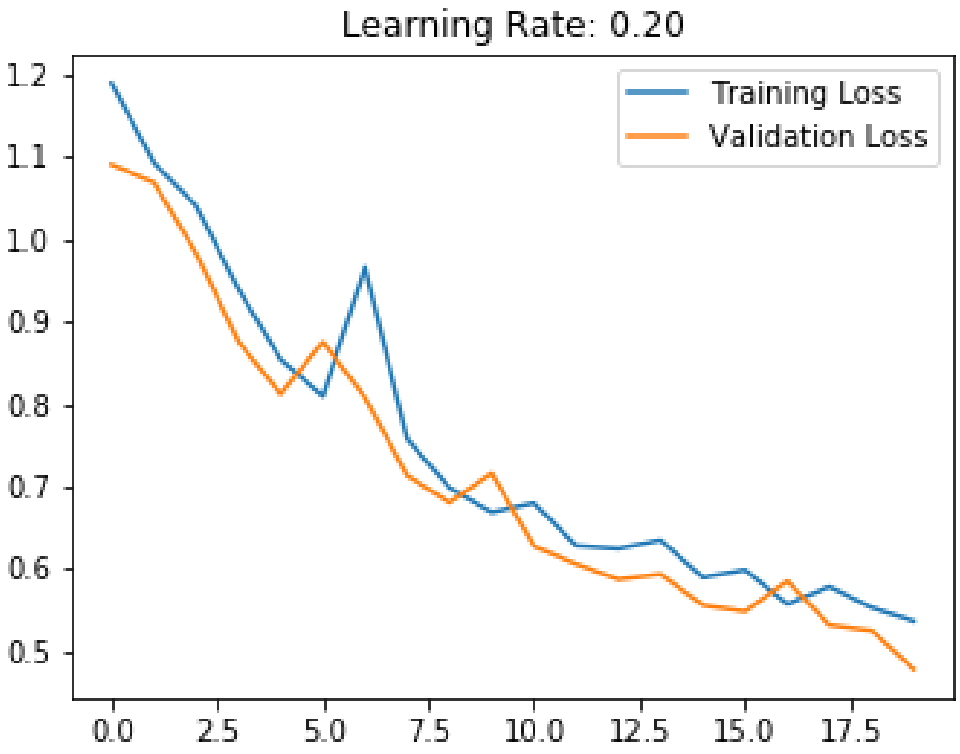
Training Frameworks:

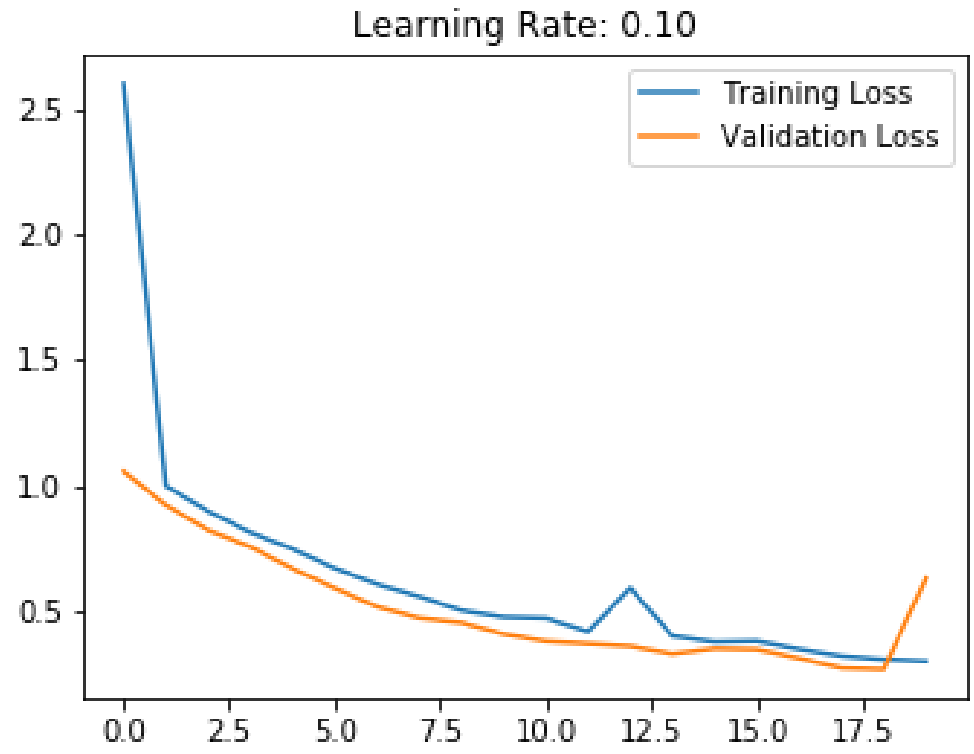
- **Huggingface Transformers**
- **PyTorch Lightning**
- **Etc**

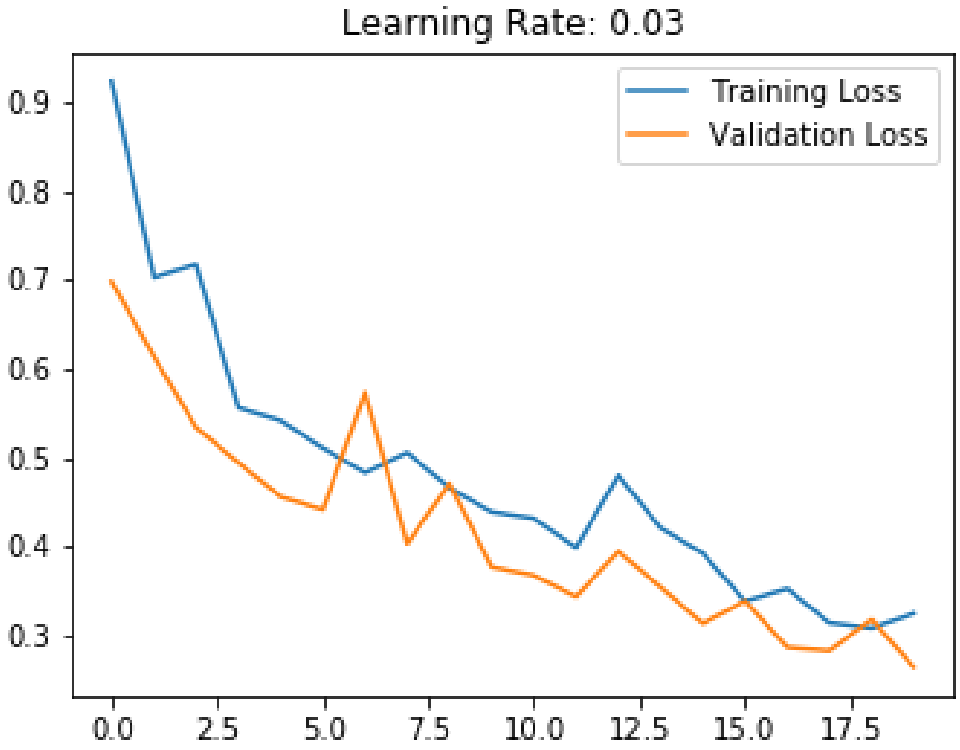
Siêu tham số (Hyperparameter) - HF Transformers

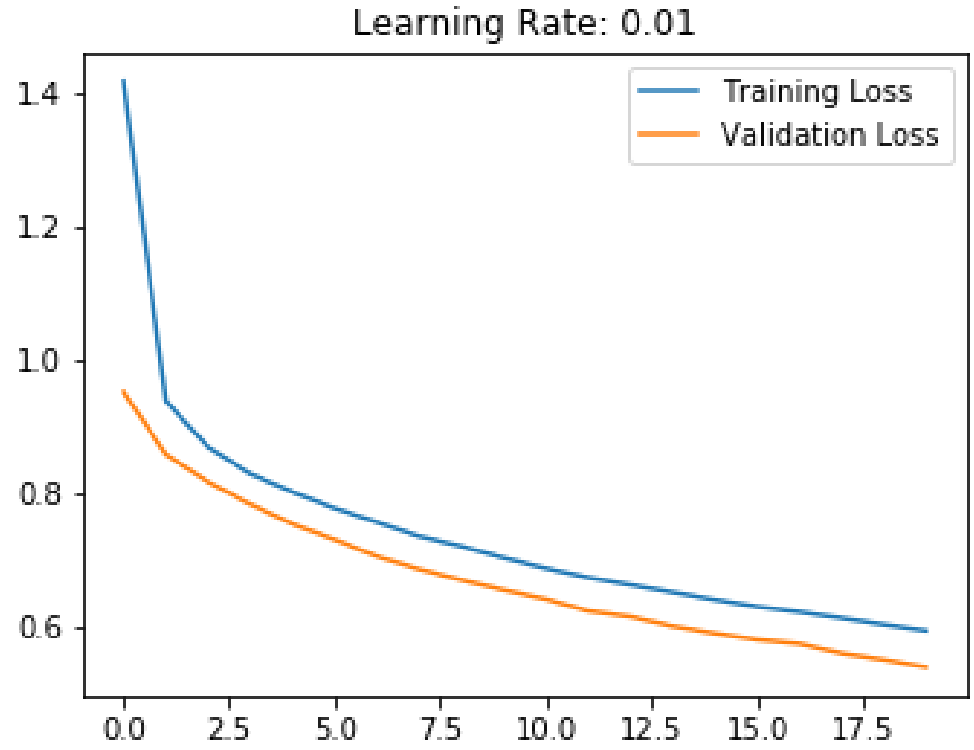
- **learning_rate:**
  - (float, optional, defaults to 5e-5) — The initial learning rate for AdamW optimizer.
- **num_train_epochs**:
  - Total number of training epochs to perform (if not an integer, will perform the decimal part percents of the last epoch before stopping training).
- **warmup_steps** :
  - (int, optional, defaults to 0) — Number of steps used for a linear warmup from 0 to learning_rate. Overrides any effect of warmup_ratio.
- **per_device_train_batch_size**:
  - The batch size per GPU/XPU/TPU/MPS/NPU core/CPU for training.
- **per_device_eval_batch_size**: The batch size per GPU/XPU/TPU/MPS/NPU core/CPU for evaluation.
- **gradient_accumulation_steps**: Number of updates steps to accumulate the gradients for, before performing a backward/update pass.

Learning Rate









DIGITAL FORTRESS

Thanks For Watching

Feel free to Contact Us

Tu Pham
Digital Fortress