Huggingface - Zero to Hero

Norapat Buppodom
Machine Learning Consultant - Thinking Machines
Super AI Engineer SS2 Gold Medal

Ecosystem

Official Libraries

Transformers	Main Machine Learning Libraries
Datasets	Download & Process Machine Learning Datasets
Accelerate	Multi-GPU Training
PEFT	Efficeint LLM training (Lora, QLora)
Optimum Documentation	General train/inference speed up
TRL Transformer Reinforcement Learning	Integrated LLM Training (PEFT+Accelerate)
pytorch-image-models	Vision AI
Diffusers	Image/Video/Audio Generation
gradio	AI Demo

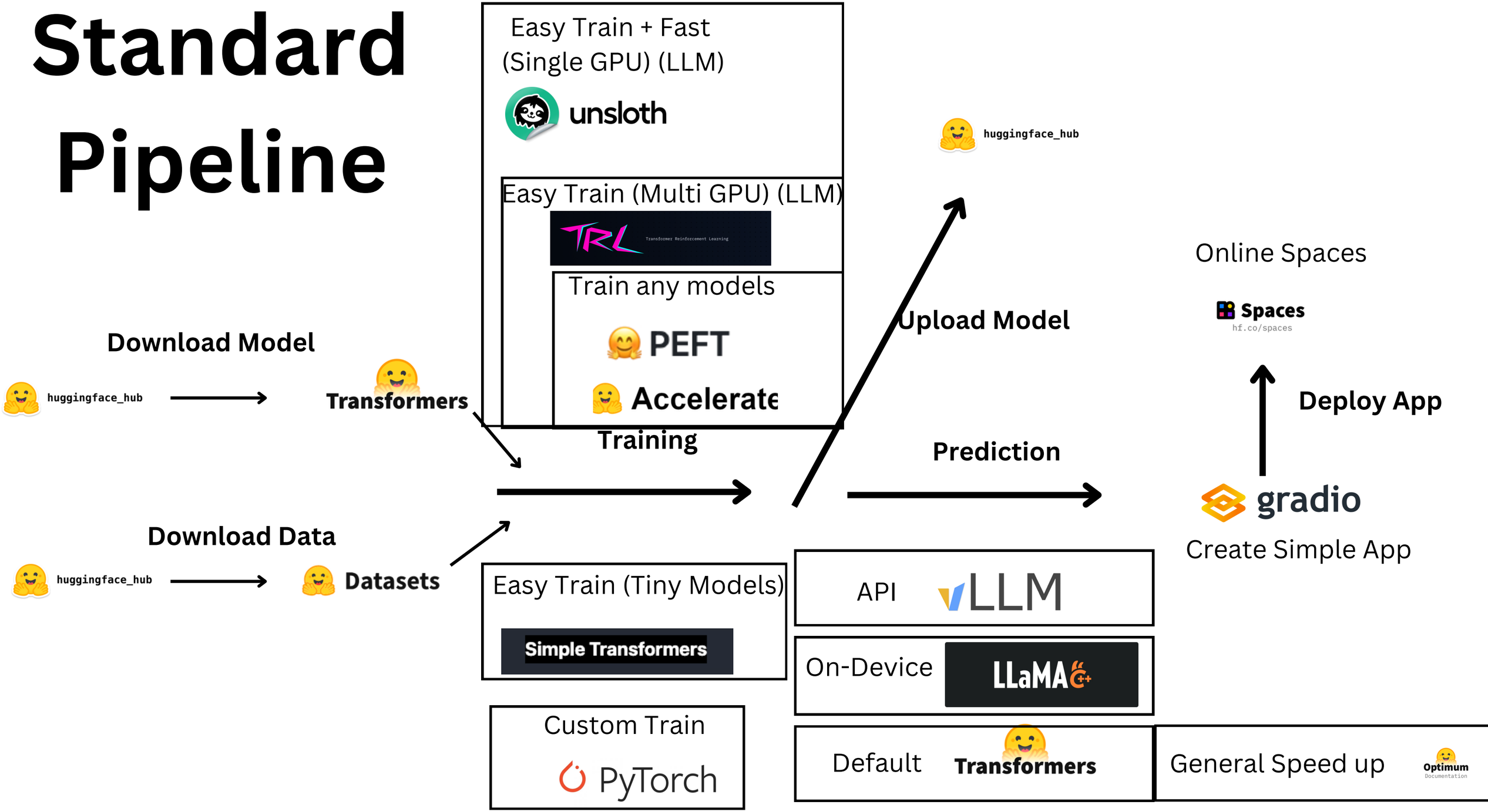
Thirdparty Libraries

unsloth	Faster TRL for single GPU
Simple Transformers	Easy Training Tiny Models
sentence-transformers	Sentence Similartiy/Search
BERTopic	Topic Modeling
vLLM	High Performance LLM Inference
LLaMA C++	On Device LLM Inference
huggingface_hub	Model/Data Hosting
Spaces hf.co/spaces	AI Demo Hosting

Standard Pipeline

Prepare Model/Data → Train Model → Prediction

Standard Pipeline



Load Models

Download Model

`huggingface_hub` → Transformers

How to find good models on Huggingface?

https://huggingface.co/models

Model Hands-on

- BERT - Masked Language Modeling
- GPT - Typhoon Chat
- CLIP - Siglip Multilingual
- OCR - openthaigpt/thai-trocr
- ASR - thonburian whisper
- Image Captioning - kkatiz/THAI-BLIP-2

Tools Covered: Transformers huggingface_hub

Train Models

Download Model

huggingface_hub → Transformers

Download Data

huggingface_hub → Datasets

Easy Train (Tiny Models)

Simple Transformers

Custom Train

PyTorch

Easy Train + Fast (Single GPU) (LLM)

unsloth

Easy Train (Multi GPU) (LLM)

TRL Transformer Reinforcement Learning

Train any models

PEFT

Accelerate

Accelerate Hands-on

- Train Sentiment Analysis Model

Tools Covered: Transformers Datasets huggingface_hub Accelerate

Simple Transformers Hands-on

Simple Transformers

- Train NER model: Thai-NER dataset

Tools Covered: Simple Transformers Datasets huggingface_hub

Unsloth

Free

Freeware of our standard version of unsloth

Get started

- Open-source
- Supports Mistral, Gemma
- Supports LLama 1, 2, 3
- Single GPU support
- Supports 4 bit, 16 bit LoRA

unsloth *Pro*

Unlock multi GPU support + 2.5x faster training + 20% less VRAM

Contact us

- 2.5x number of GPUs faster than FA2
- 20% less memory than OSS
- Multi GPU support
- Up to 8 GPUS support
- For any usecase

GET STARTED

Unsloth Notebooks

Below is a list of all our notebooks:

• Google Colab • Kaggle

Main notebooks:

- Llama 3.1 (8B) - GRPO reasoning
- Phi-4 (14B) - GRPO reasoning
- Qwen2.5 (3B) - GRPO reasoning
- Phi-4 (14B)
- Llama 3.1 (8B)
- Llama 3.2 (1B + 3B)
- Mistral v0.3 Instruct (7B)
- Gemma 2 (9B)
- Qwen 2.5 (7B)

Vision notebooks:

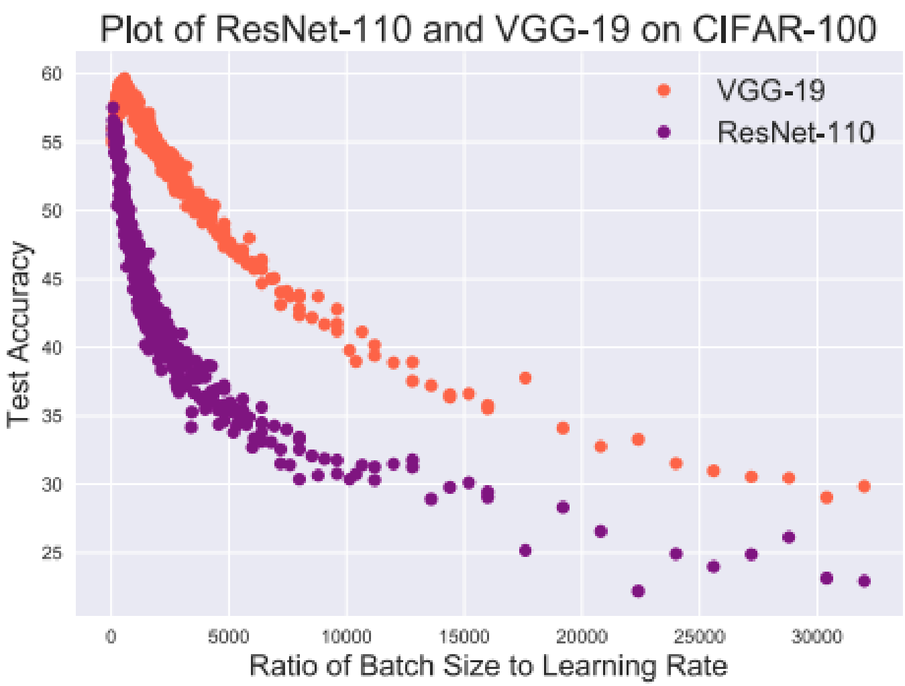
- Llama 3.2 Vision (11B)
- Qwen2-VL (7B)
- Pixtral (12B) 2409

“Training LLM on Single GPU”

Quick Fine-tuning Guide

- Learning Rate ⬆
- Batch Size ⬆

- Learning Rate ⬇
- Batch Size ⬇



Control Batch Size and Learning Rate to Generalize Well: Theoretical and Empirical Evidence: NeuralIPS 2019

Quick Fine-tuning Guide 2

Base vs Instruct Model

Should I Choose Instruct or Base?

The decision often depends on the quantity, quality, and type of your data:

- **1,000+ Rows of Data**: If you have a large dataset with over 1,000 rows, it's generally best to fine-tune the base model.
- **300–1,000 Rows of High-Quality Data**: With a medium-sized, high-quality dataset, fine-tuning the base or instruct model are both viable options.
- **Less than 300 Rows**: For smaller datasets, the instruct model is typically the better choice. Fine-tuning the instruct model enables it to align with specific needs while preserving its built-in instructional capabilities. This ensures it can follow general instructions without additional input unless you intend to significantly alter its functionality.
- For information how how big your dataset should be, see here

https://docs.unsloth.ai/get-started/beginner-start-here/what-model-should-i-use

Base vs Instruct Model

Quick Fine-tuning Guide 2

Base Llama

```python
import torch
from transformers import pipeline

model_id = "meta-llama/Llama-3.2-3B"

pipe = pipeline(
    "text-generation",
    model=model_id,
    torch_dtype=torch.bfloat16,
    device_map="auto"
)

pipe("The key to life is")
```

https://huggingface.co/meta-llama/Llama-3.2-3B

Instruct Llama

```python
import torch
from transformers import pipeline

model_id = "meta-llama/Llama-3.2-3B-Instruct"
pipe = pipeline(
    "text-generation",
    model=model_id,
    torch_dtype=torch.bfloat16,
    device_map="auto",
)
messages = [
    {"role": "system", "content": "You are a pirate chatbot who always responds i
    {"role": "user", "content": "Who are you?"},
]
outputs = pipe(
    messages,
    max_new_tokens=256,
)
print(outputs[0]["generated_text"][-1])
```

https://huggingface.co/meta-llama/Llama-3.2-3B-Instruct

Quick Fine-tuning Guide 3

	Full-finetuning	Lora	QLora
Params	100%	0.1-1%	0.1-1%
Precisions	32-16 bit	32-16 bit	4 bit
Memory	💵💵💵💵💵	💵💵💵	💵
Data Requirements	💵💵💵💵💵	💵💵💵	💵💵💵
Speed	🔥	🔥🔥🔥	🔥🔥
Accuracy	👍👍👍	👍👍	👍

Quick Fine-tuning Guide 3

Model parameters	QLoRA (4-bit) VRAM	LoRA (16-bit) VRAM
3B	3.5 GB	8 GB
7B	5 GB	19 GB
8B	6 GB	22 GB
9B	6.5 GB	24 GB
11B	7.5 GB	29 GB
14B	8.5 GB	33 GB
27B	16GB	64GB
32B	19 GB	76 GB
40B	24GB	96GB
70B	41 GB	164 GB
81B	48GB	192GB
90B	53GB	212GB
405B	237 GB	950 GB

https://docs.unsloth.ai/get-started/beginner-start-here/unsloth-requirements

Unsloth Hands-on

unsloth

- Train LLM Text2SQL: **Llama3.2-3B**
- Train VLM Image Captioning: **Qwen2.5-Instruct**

Tools Covered: unsloth TRL Transformer Reinforcement Learning Datasets PEFT huggingface_hub

Prediction

→

API	vLLM
On-Device	LLaMA C++
Default	Transformers
General Speed up	Optimum Documentation

Optinum Hands-on

Optimum
Documentation

- Run Language Detection faster with ONNX runtime

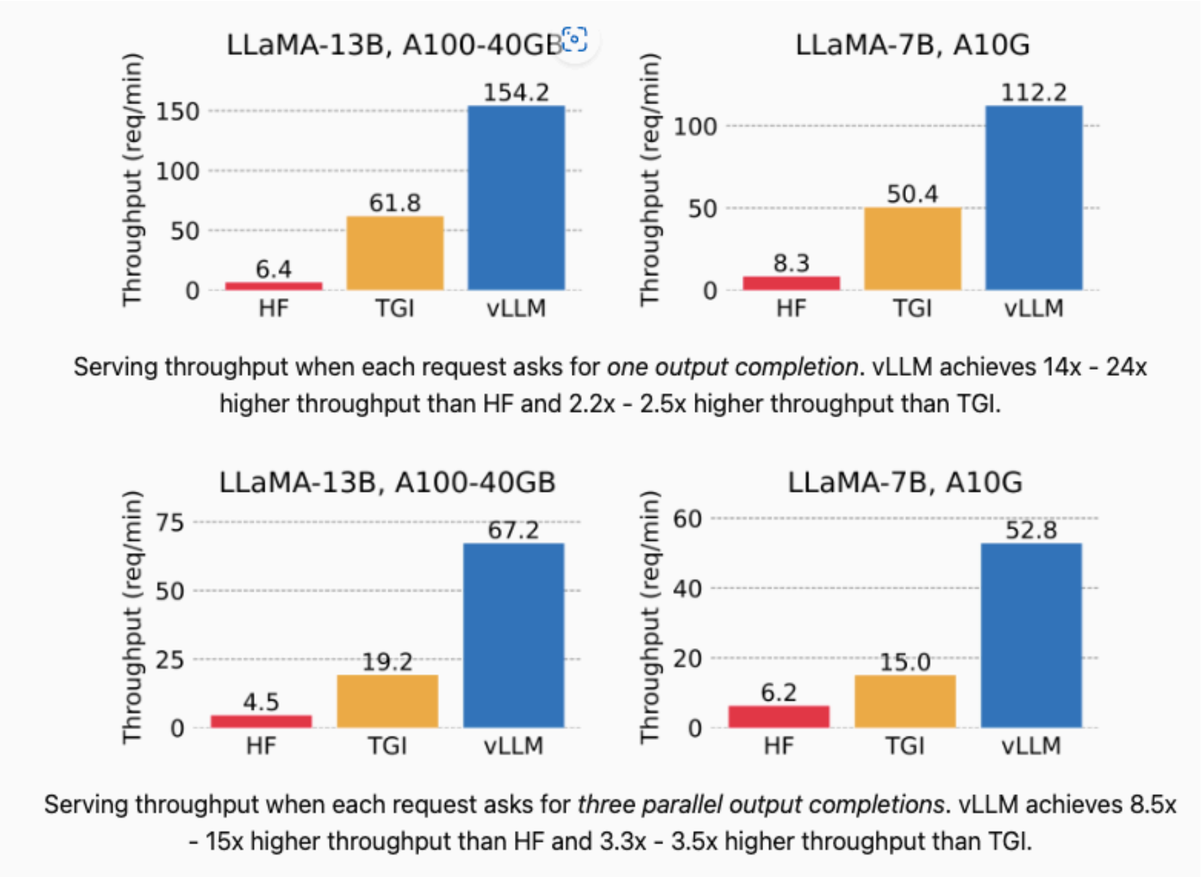
Tools Covered: Optimum Documentation ONNX

vLLM

vLLM

Good for:

- You want parallel prediction
- You want API
- You have ‘fast’ gpu



Llama.cpp

Good for:

- You want to run LLM on local environment
- You want to run LLM on consumer hardware

VLLM Hands-on

- Train LLM Text2SQL: **Run Llama3.3 3B Finetuned Text2Sql on Server Grade GPU**
- Train VLM Image Caption: **Run Qwen-VL 2B Finetuned COCO Thai on Server Grade GPU**

Tools Covered: vLLM huggingface_hub Transformers

Llama.cpp Hands-on

Jan

https://jan.ai/

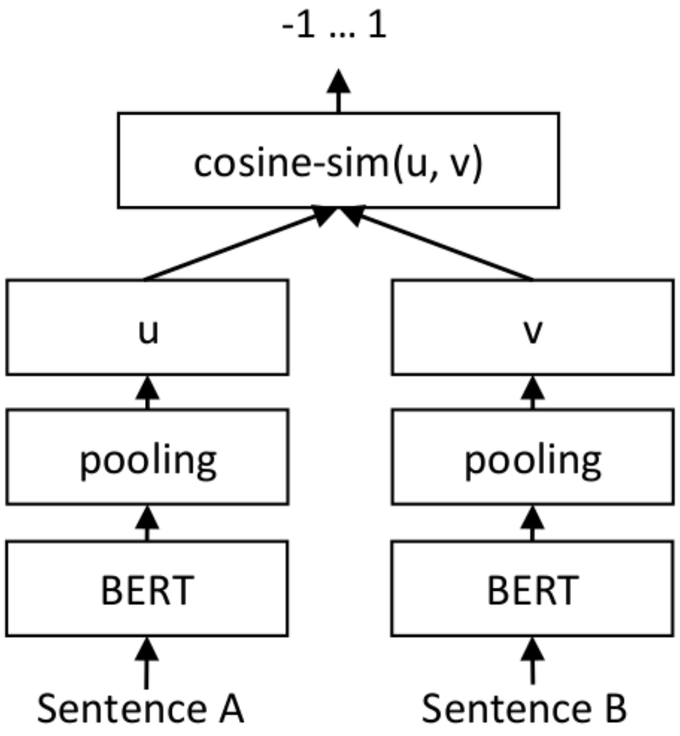
- Train LLM Text2SQL: **Run Llama3.3 3B Finetuned Text2Sql on Local Device**

Tools Covered: LLaMA C++

Sentence Transformers

Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks

EMNLP 2019



How to choose good sentence transformers model in Thai?

https://github.com/mrperat/Thai-Sentence-Vector-Benchmark

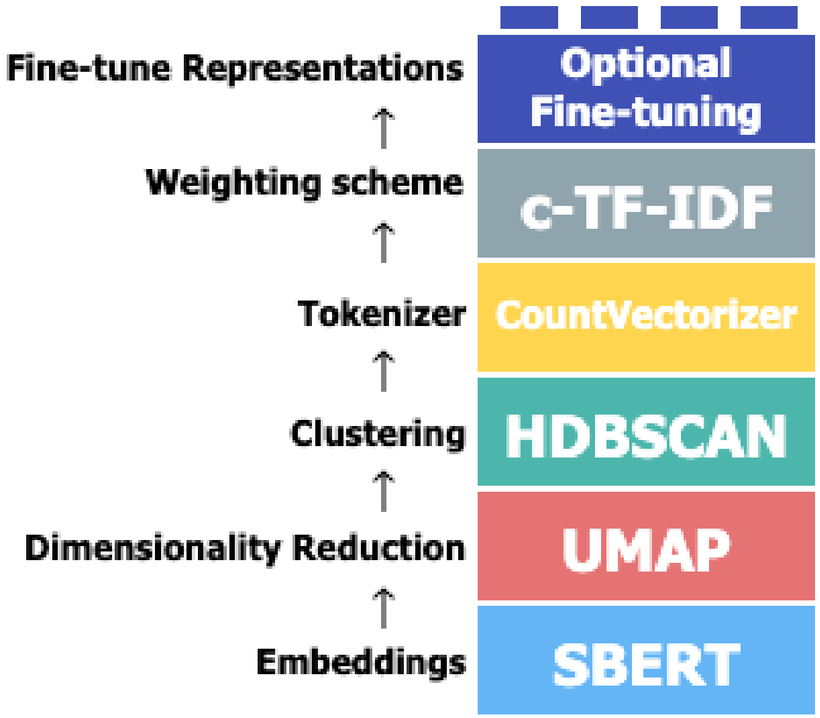
Sentence Transformers Hands-on

- Demo BGE-m3 with iapp wiki datset

Tools Covered: sentence-transformers huggingface_hub Datasets NumPy

BertTopic

“State of the art topic modeling”



BertTopic Hands-on

- Topic Modeling on Thai-wikipedia

Tools Covered: BERTopic, sentence-transformers, huggingface_hub, Datasets, PyThaiNLP (We build Thai NLP.)

Thank You