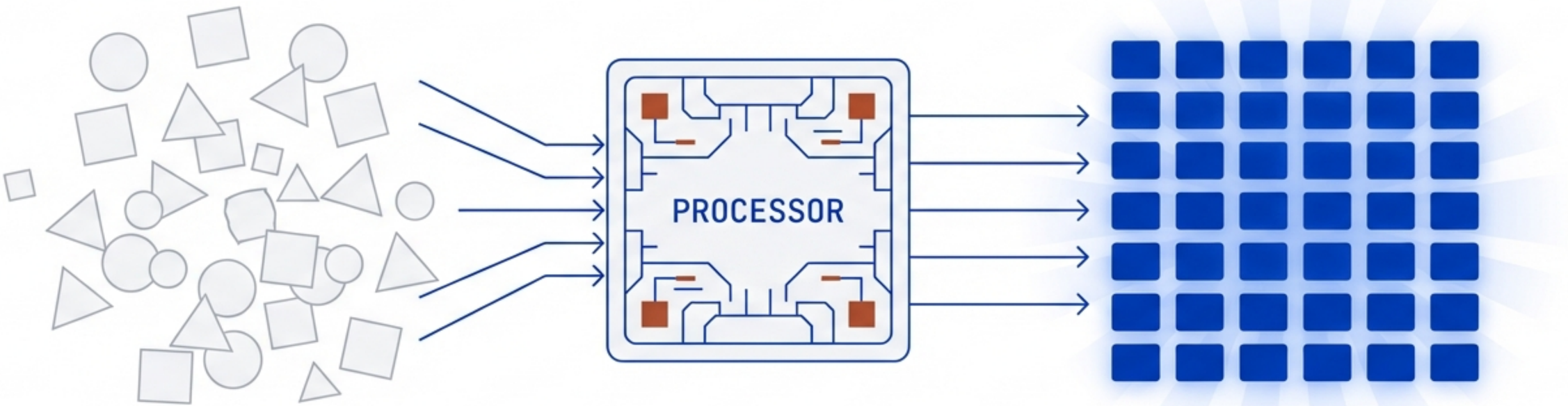
Unsloth Datasets Guide

Mastering Data Preparation for Fine-Tuning



Documentation Series

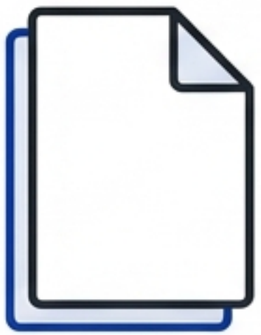
UNSLOTH

The Anatomy of a Dataset

LLMs do not read; they calculate. The journey from text to tensors.

Stage 1

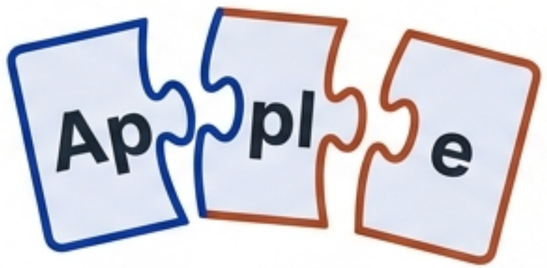
Raw Text



"Apple"

Stage 2

Tokenisation

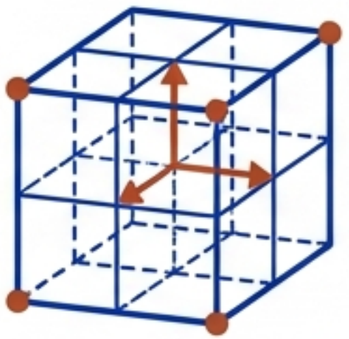


Breaks text into sub-words (tokens)

[142, 294]

Stage 3

Embeddings



Tokens converted to numerical vectors

[0.12, -0.45, 0.88...]

Note:
The Critical Link: The **Chat Template** dictates how the tokenizer structures this conversation for the model.

The Data Strategy Blueprint

Before writing code, define the three **pillars** of your dataset strategy.

1. Goal Definition

- **Task Adaptation** (Summarisation, Classification)
- **Roleplay** (Character emulation)
- **Domain Specific** (Medical, Finance)

2. Output Format

- **Structure**: JSON, HTML, Code, Plain Text
- **Language**: English, Spanish, German

3. Data Origin

- **Hugging Face / Wikipedia** (Generalised)
- **Local Files** (CSV, PDF, Internal Docs)
- **Synthetic** (Generated via Local LLMs)

The Four Archetypes of Data Formats

Archetype Label	Icon	Description	Use Case
Raw Corpus		Unstructured text from books, websites, or articles.	`Continued Pretraining (CPT)` - Preserves natural flow.
Instruct		Single-turn instruction with a specific output target.	`Supervised Fine-Tuning (SFT)` - Task specific.
Conversation		Multi-turn dialogue between User and Assistant.	SFT - Chat-based assistants (**ShareGPT**, **ChatML**).
RLHF		Conversations with ranked responses for preference learning.	**Reinforcement Learning (RL).**

Engineering Editorial

The Unsloth Formatting Workflow

Standard process for ChatML datasets.

1.

Check Support

View available templates.

```
unsloth.chat_templates
```

2.

Get Template

Apply template to tokenizer.

```
get_chat_template(t
okenizer, 'llama-3')
```

Fixes base model errors.

3.

Define Function

Create formatting function.

```
formatting_prompts_func
```

4.

Map

Apply to dataset.

```
dataset.map(...,
batched=True)
```

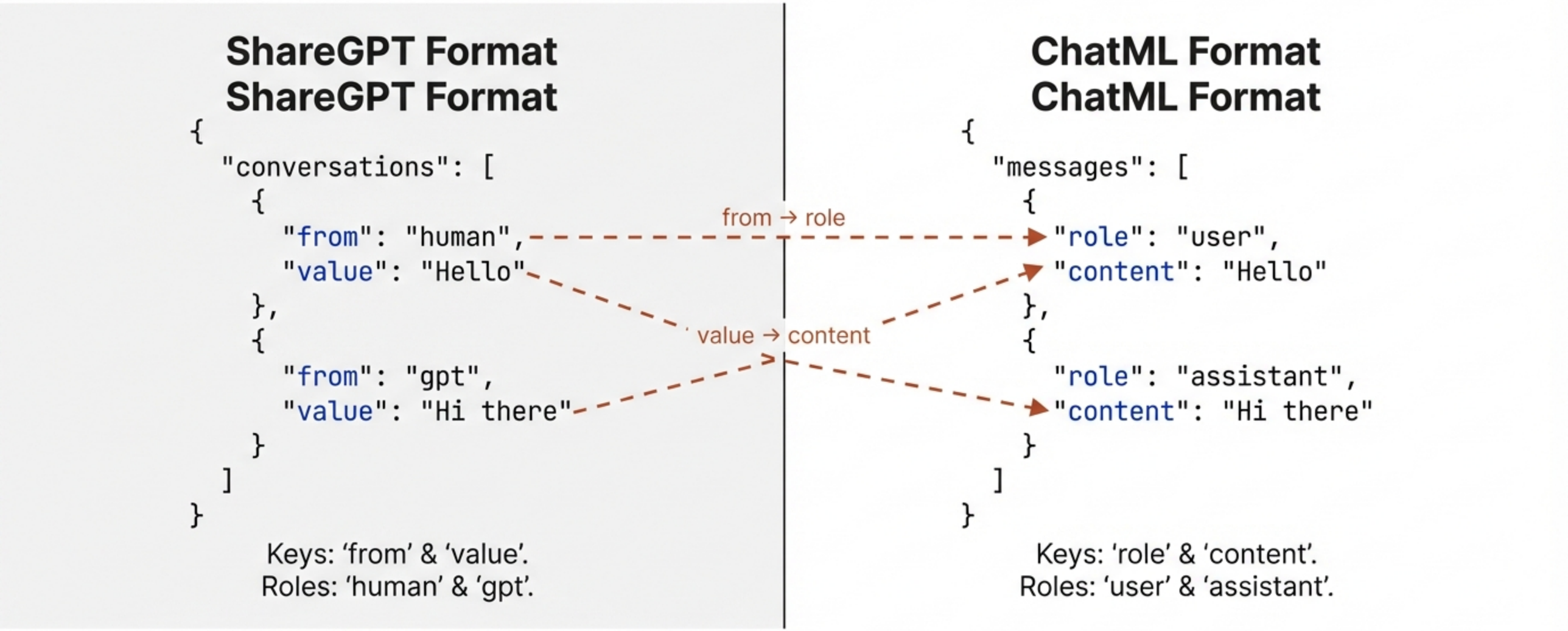
Decoding Chat Templates: ShareGPT vs. ChatML

ShareGPT Format

ShareGPT Format

```
{
  "conversations": [
    {
      "from": "human",
      "value": "Hello"
    },
    {
      "from": "gpt",
      "value": "Hi there"
    }
  ]
}
```

Keys: ‘from’ & ‘value’.
Roles: ‘human’ & ‘gpt’.



ChatML Format

ChatML Format

```
{
  "messages": [
    {
      "role": "user",
      "content": "Hello"
    },
    {
      "role": "assistant",
      "content": "Hi there"
    }
  ]
}
```

Keys: ‘role’ & ‘content’.
Roles: ‘user’ & ‘assistant’.

Default format for Hugging Face.

The Standardisation Toolkit

Converting ShareGPT to ChatML automatically.

Input: ShareGPT Format (`from/value`)

```
{"from": "human", "value", "value": "Hello", {"from": "gpt", "value": {...}}
```

standardize_sharegpt(dataset)

Output: ChatML Format (`role/content`)

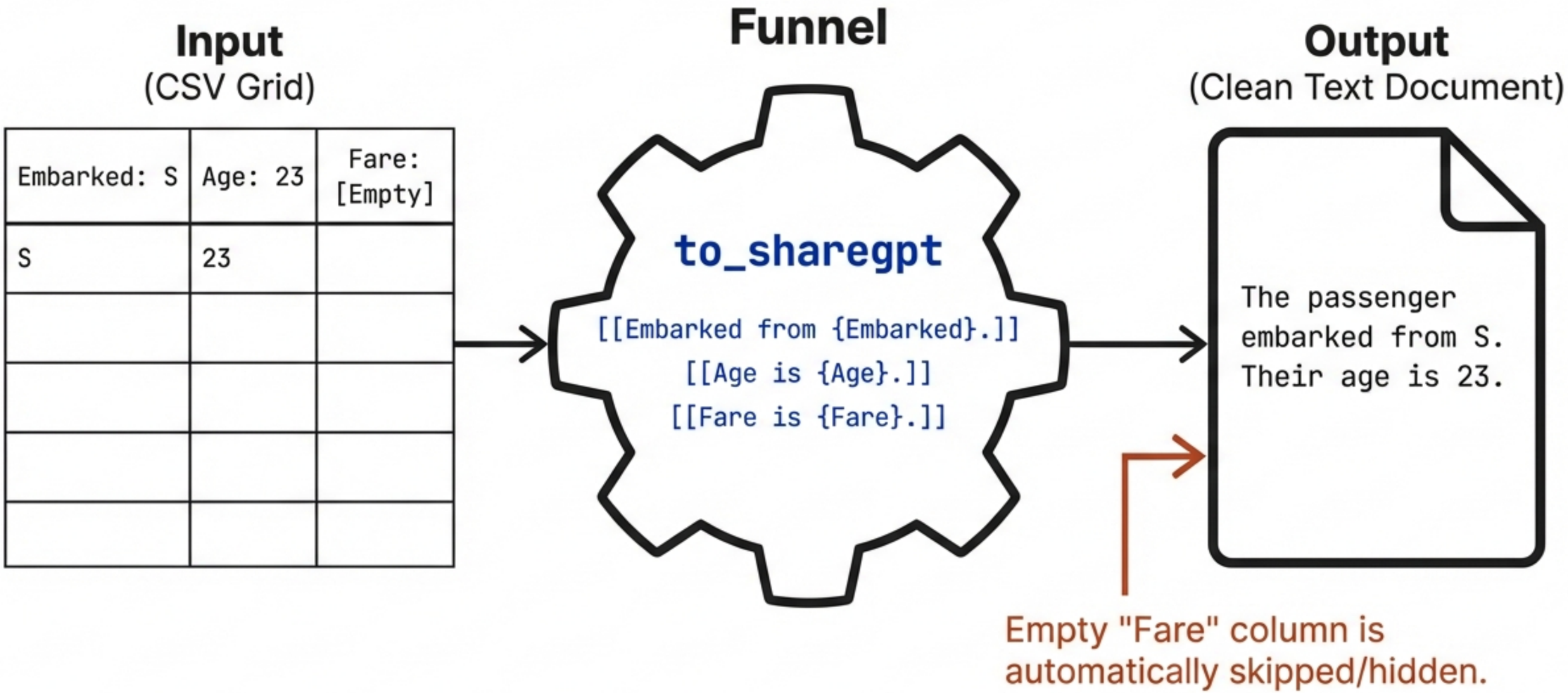
```
{"role": "user", "gpt, "assistant", "assistant", "Hello ..., "content": "Hi there"}
```

```
from unsloth.chat_templates import standardize_sharegpt

dataset = load_dataset("mlabonne/FineTome-100k", split="train")
dataset = standardize_sharegpt(dataset) # The Magic Step
dataset = dataset.map(formatting_prompts_func, batched=True)
```

Wrangling Columns: The ‘Titanic’ Problem

Merging complex tabular data into a single prompt.



From Monologue to Dialogue

Simulating multi-turn conversations from single-turn instruction datasets.

Alpaca Row 1

```
{
  "instruction": "Summarize the article.",
  "output": "The article discusses..."
}
```

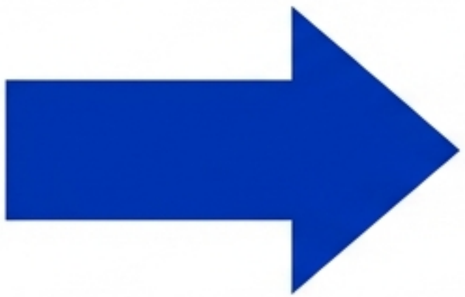
Alpaca Row 5

```
{
  "instruction": "Translate to Spanish.",
  "output": "Traduce al español..."
}
```

Alpaca Row 9

```
{
  "instruction": "Identify the main character.",
  "output": "The main character is..."
}
```

`conversation_extension = 3`



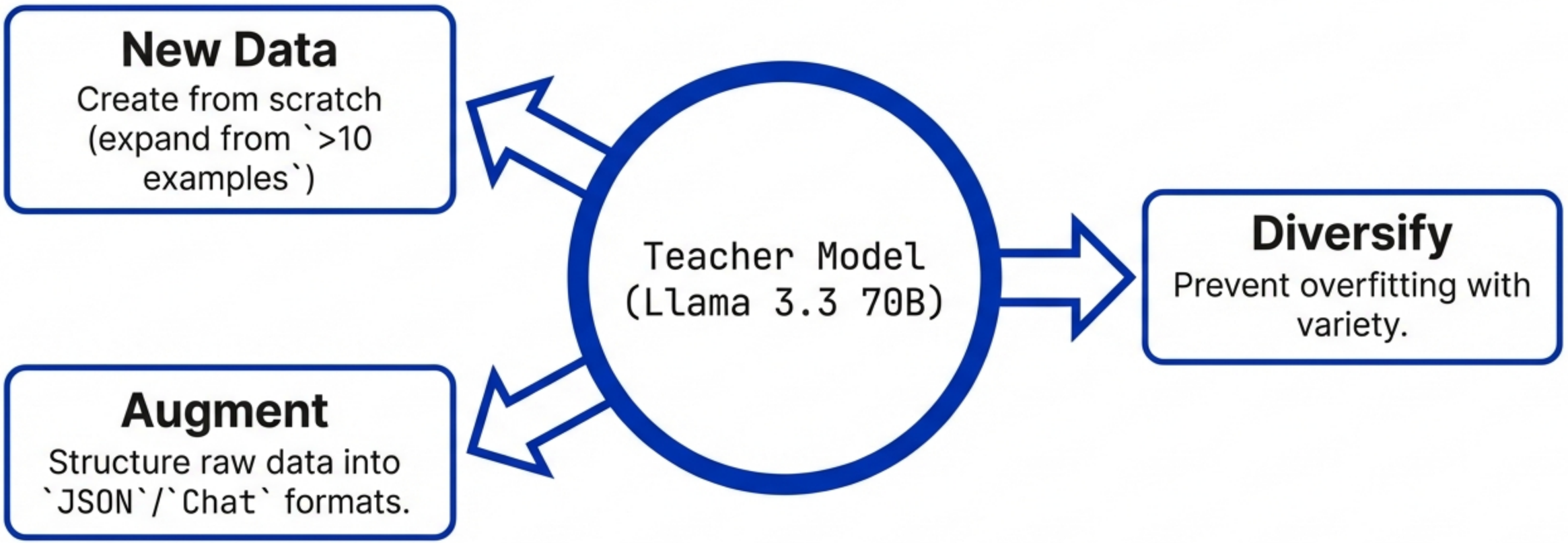
Synthetic Context

```
{
  "from": "human",
  "value": "Summarize the article."
}
{
  "from": "gpt",
  "value": "The article discusses..."
}
{
  "from": "human",
  "value": "Translate to Spanish."
}
{
  "from": "gpt",
  "value": "Traduce al español..."
}
{
  "from": "human",
  "value": "Identify the main character."
}
{
  "from": "gpt",
  "value": "The main character is..."
}
```

Workflow: `to_sharegpt` -> `conversation_extension` -> `standardize_sharegpt`

Synthetic Data Generation

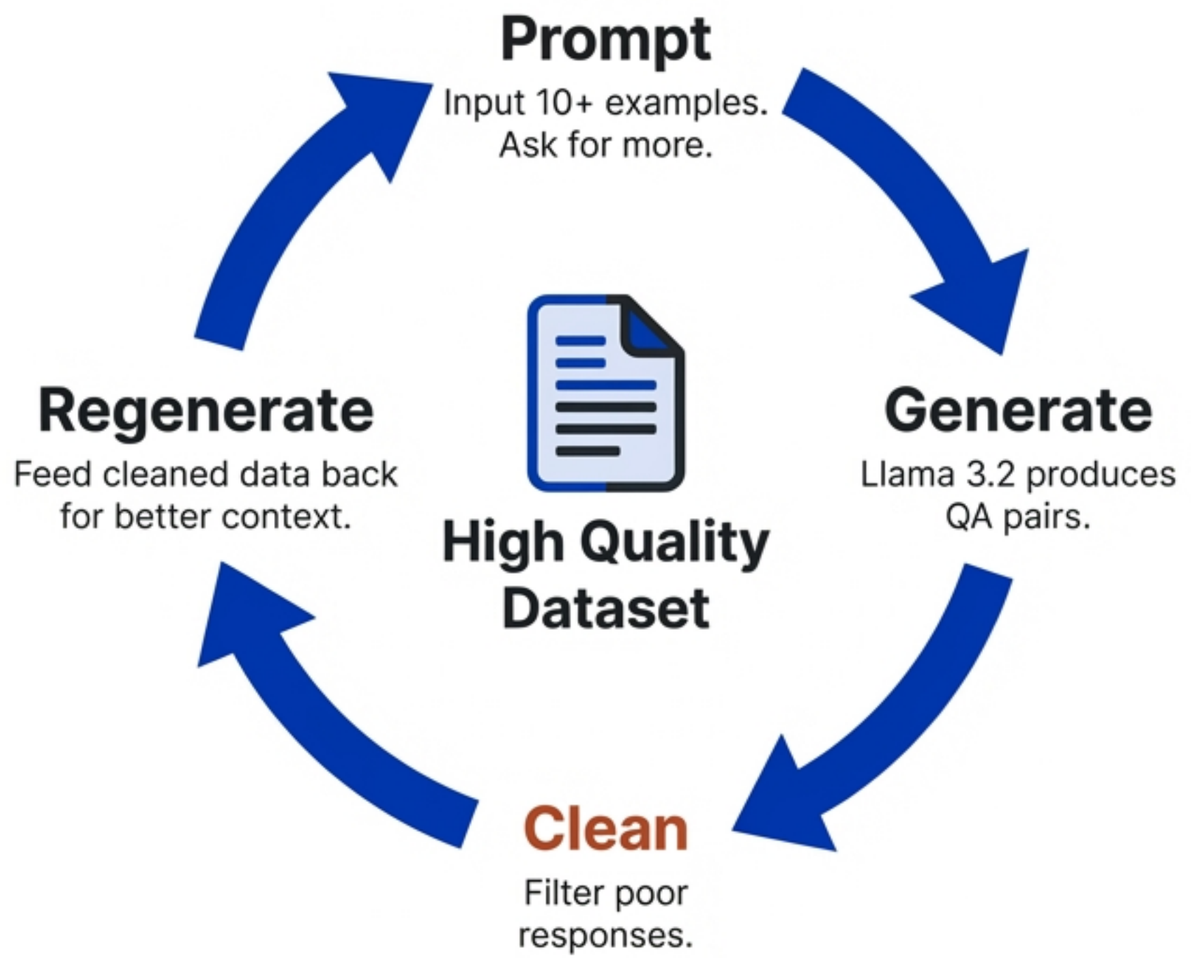
Using local LLMs to solve data scarcity.



Tools: `vLLM, Ollama, llama.cpp`

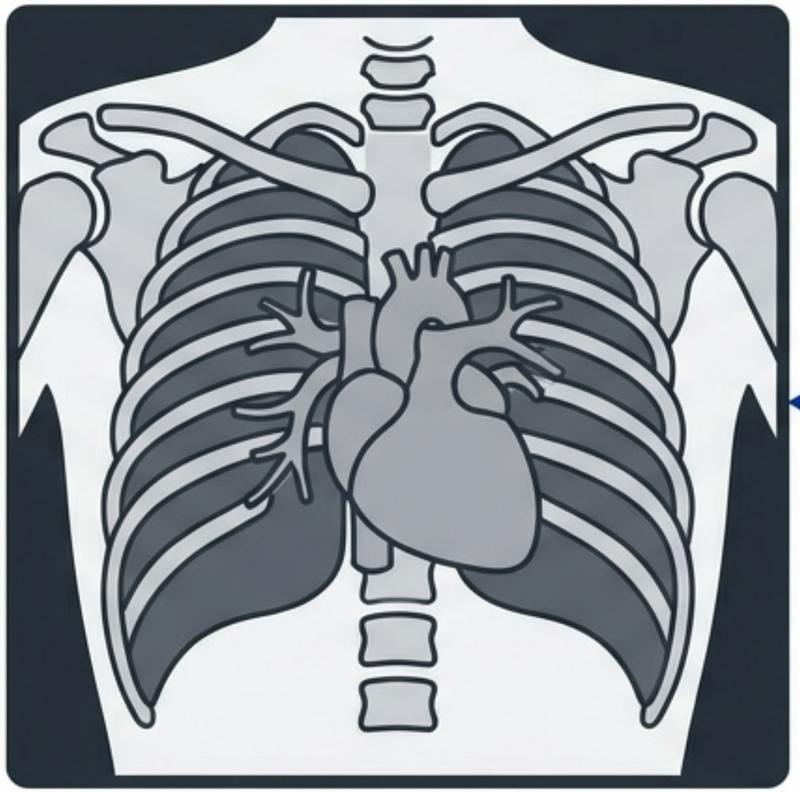
The Synthetic Workflow Loop

Unsloth & Meta Synthetic Dataset Notebook



Key Prompt: "Prompt for generating more dialogue on an existing dataset."

Vision Fine-Tuning: Multimodal Formatting



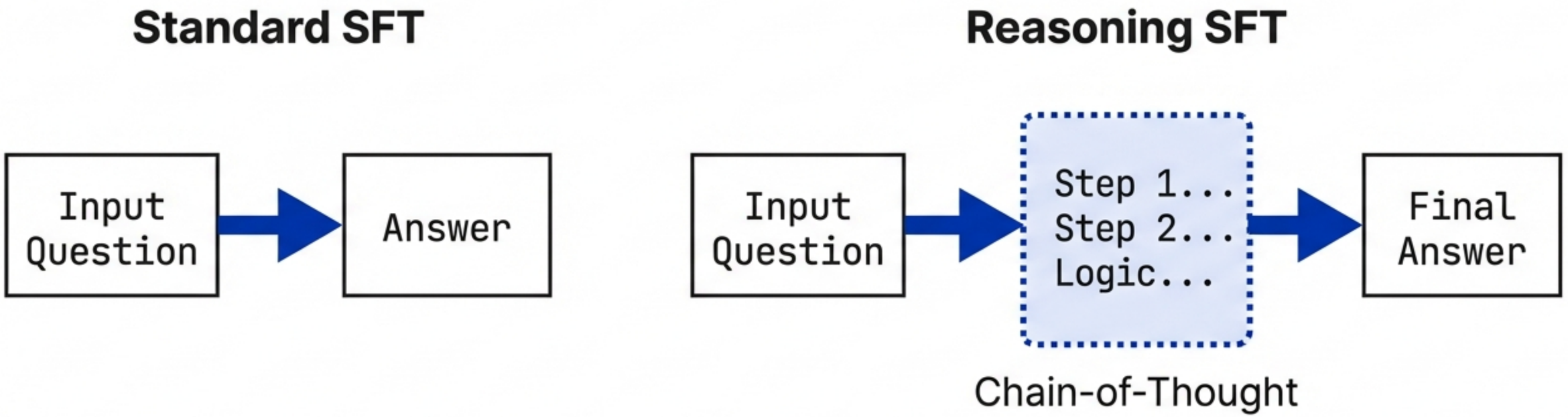
Input Image

```
{
  "role": "user",
  "content": [
    {
      "type": "text",
      "text": "Describe this image." },
    },
    { "type": "image",
    "image": "<PIL Image Object>" }
    }
  ]
}
```

Case Study: ROCO Radiography Dataset.

Specialised Tuning: Reasoning & R1

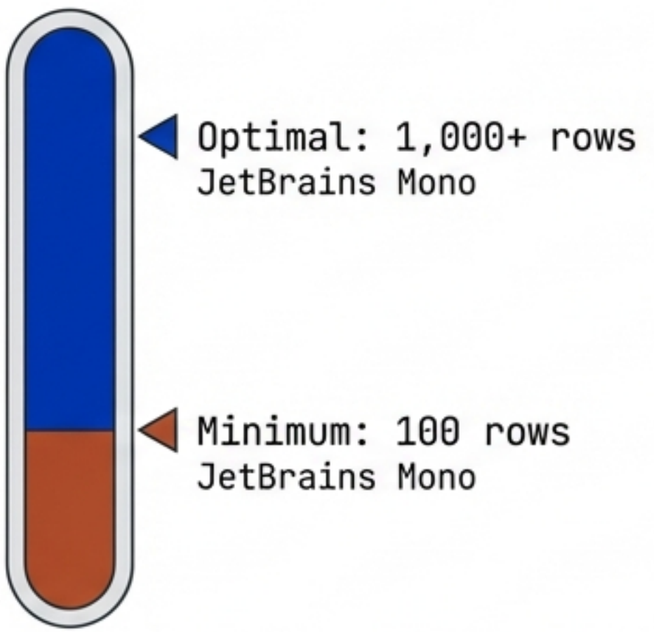
Teaching models *how* to think, not just what to answer.



For DeepSeek-R1 style models, the dataset must explicitly include the reasoning steps in the assistant response.

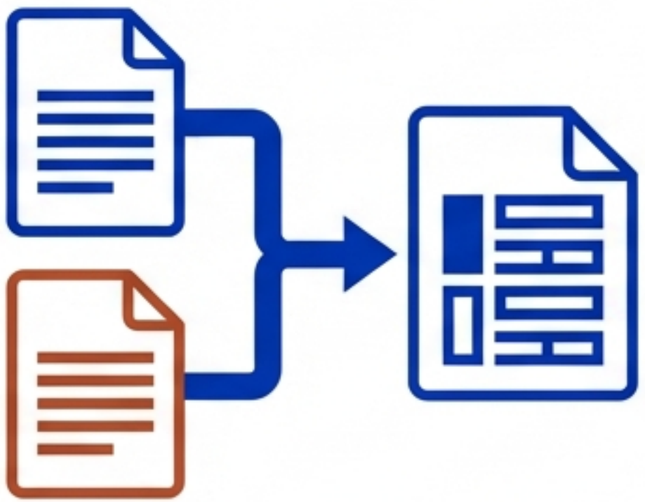
Optimisation & Best Practices

Dataset Size



Quality > Quantity.

Multiple Datasets



Standardise formats and merge into one file before training.

Refining Models

Avoid sequential fine-tuning. Always combine new data with old data and retrain from base.

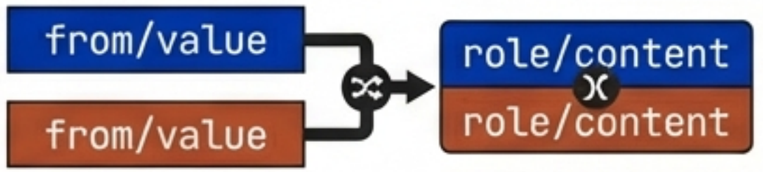
The Unsloth Cheat Sheet

Fix

`get_chat_template`

Applies correct template & fixes base model errors.

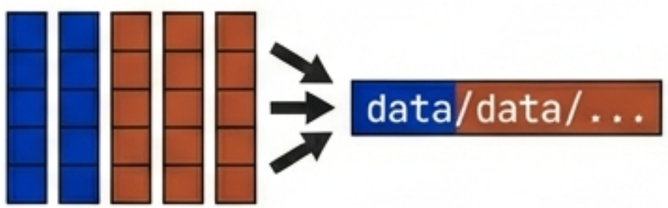
Standardise



`standardize_sharegpt`

Converts `from/value` (ShareGPT) to `role/content` (ChatML).

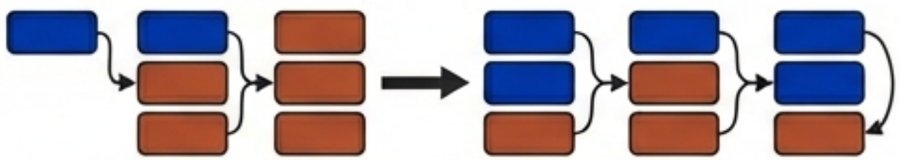
Merge



`to_sharegpt`

Collapses CSV columns into a single prompt string.

Expand



`conversation_extension`

Stitches single-turn rows into multi-turn dialogues.

Structure is the signal in the noise.