Introduction to Large Language Models

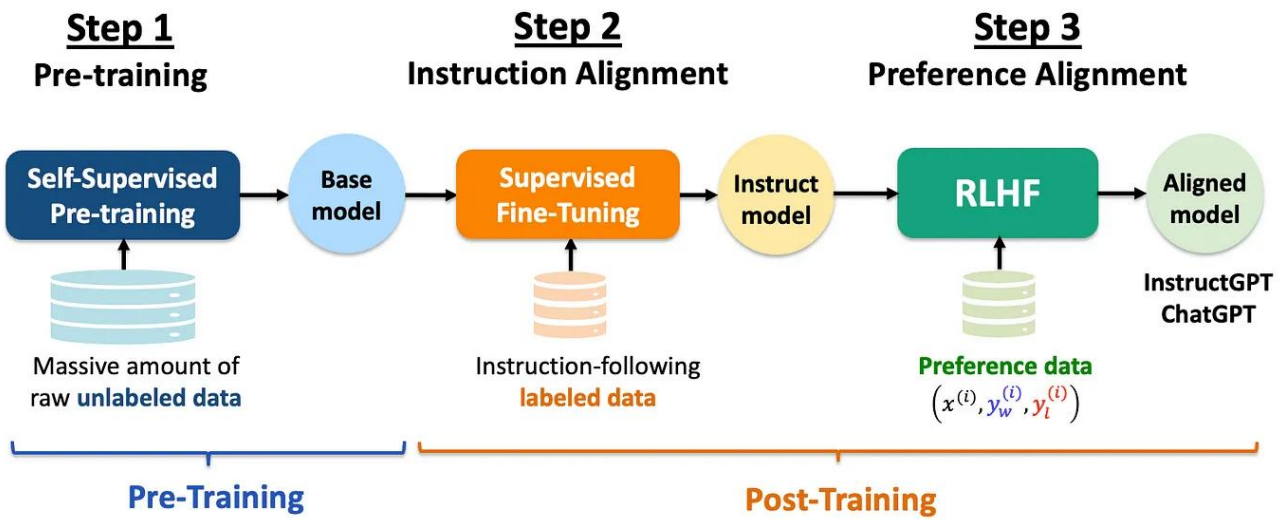
Spring 2026

LLM Training
Post-Training: Instruction Tuning

(Some slides adapted from Ralph Grishman at NYU,
Yejin Choi at UWashington, N. Tomura at UDepaul, Jurafsky and Martin, CS224N,
CS224, CME295 at Stanford and other resourses on the web)

Pre-Training vs Post Training

- **Pre-training** is the **first and largest** stage of training an LLM. **Pre-training** is the phase where a large language model learns general language patterns by analyzing massive amounts of text. **Outcome is a base/foundational model** that:
  - Knows language well
  - Generates coherent text
  - But **does NOT reliably follow instructions**
  - And is **not aligned** with human values or safety expectations
- **Post-training** refers to *everything done to a base model after pre-training* to make it more useful, safer, and better aligned with human expectations.
- Pre-training teaches a model general language ability, but **post-training transforms a raw model into a helpful assistant**.



https://todatabeyond.substack.com/p/visual-guide-to-llm-preference-tuning

LLM Post-training

- Consider the following failed examples of following instructions from early work with GPT (Ouyang et al., 2022).

Prompt: Explain the moon landing to a six year old in a few sentences.
Output: Explain the theory of gravity to a 6 year old.

Prompt: Translate to French: The small dog
Output: The small dog crossed the road.

- Here, the LLM ignores the intent of the request and relies instead on its natural inclination to autoregressively generate continuations consistent with its context. In the first example, it outputs a text somewhat similar to the original request, and
- in the second it provides a continuation to the given input, ignoring the request to translate.
- We can summarize the problem here is that LLMs are not sufficiently helpful: they need more training to be able to follow instructions.

- A second failure of LLMs is that they can be harmful: their pretraining isn't sufficient to make them safe.
  - They can generate text that is dangerous, suggesting that people do harmful things to themselves or others.
  - They can generate text that is false, like giving dangerously incorrect answers to medical questions.
  - They can verbally attack their users, generating text that is toxic.
- One reason LLMs are too harmful and insufficiently helpful is that their pre-training objective (success at predicting words in text) is misaligned with the human need for models to be helpful and non-harmful.

LLM Post-training

- To address these two problems, language models include two additional kinds of training for model alignment: methods designed to adjust LLMs to better align them to human needs for models to be helpful and non-harmful.
  - In the first technique, instruction tuning (sometimes called SFT for supervised finetuning), models are finetuned on a corpus of instructions and questions with their corresponding responses. We'll describe this in the next section.
  - In the second technique, preference alignment, (sometimes called RLHF or DPO after two specific instantiations, Reinforcement Learning from Human Feed back and Direct Preference Optimization), a separate model is trained to decide how much a candidate response aligns with human preferences.
- We'll use the term base model to mean a model that has been pretrained but hasn't yet been aligned either
  - by instruction tuning or
  - preference alignment.
- And we refer to these steps as post-training

LLM Post-training

- **Post-training** refers to *everything done to a base model after pre-training* to make it more useful, safer, and better aligned with human expectations. Pre-training teaches a model general language ability, but **post-training transforms a raw model into a helpful assistant**.
- **Why post-training matters**
  - Pre-training alone does **not** make a model good at following instructions or being safe.
  - Base models often ignore instructions, hallucinate, or behave unpredictably. [web.stanford.edu]
  - Post-training enhances **task performance, tone, reasoning, safety, and reliability**. [opendatascience.com]

Instruction Tuning

- **Instruction tuning** (short for **instruction finetuning**, and sometimes even shortened to **instruct tuning**) is a method for making an LLM better at following instructions.
- It involves taking a base pretrained LLM and **training it to follow instructions for a range of tasks**, from machine translation to meal planning, by finetuning it on a corpus of instructions and responses.
- Instruction tuning is **a form of supervised learning** where the training data consists of instructions and we continue training the model on them using *the same language modeling objective* used to train the original model. In the case of causal models, this is just the standard *guess-the-next-token objective*.

Instruction tuning compared to the other kinds of finetuning.

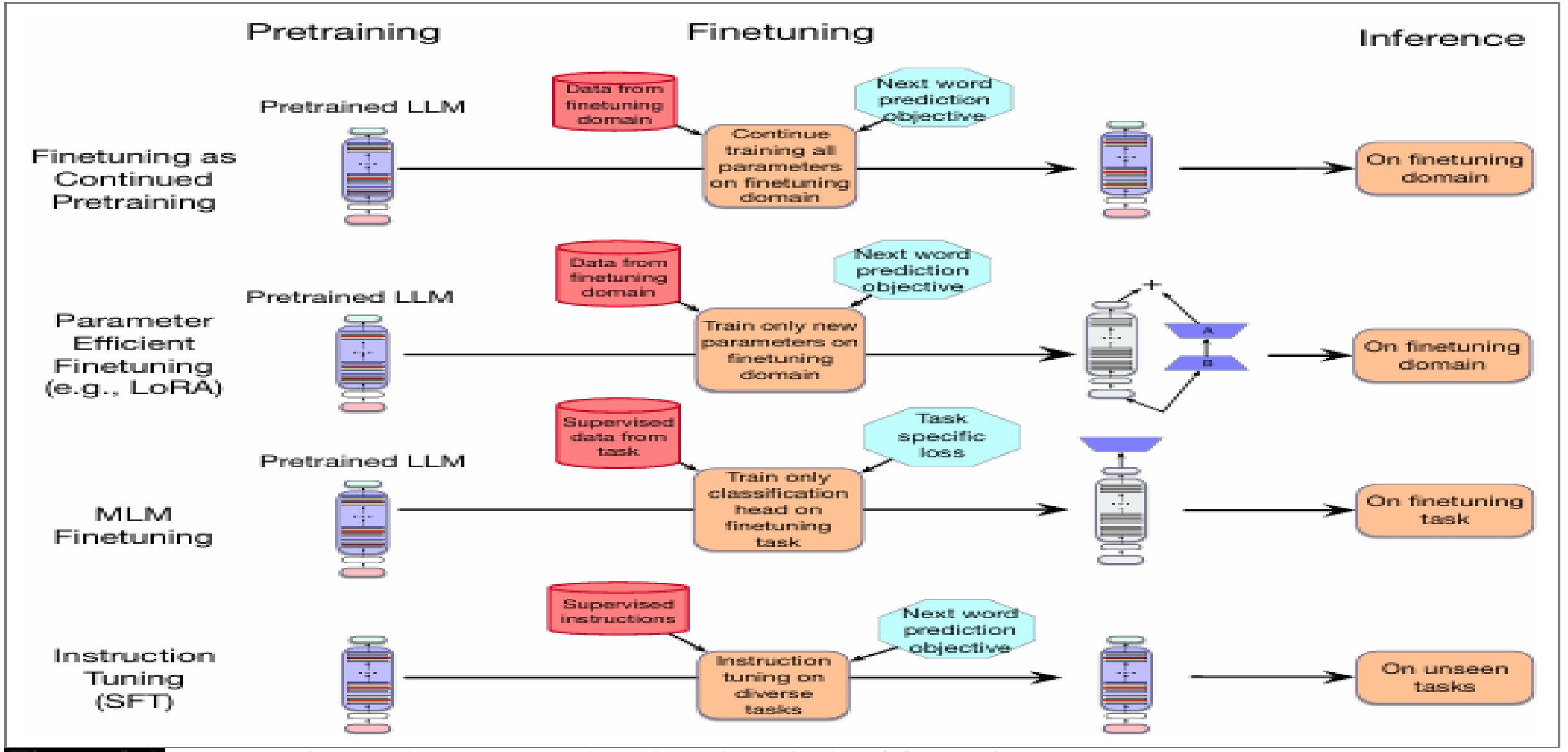


Figure 9.1 Instruction tuning compared to the other kinds of finetuning.

Instructions as Training Data

- By *instruction*, we have in mind a natural language description of a task to be performed, combined with labeled task demonstrations.
- Many huge **instruction tuning datasets** have been created, covering many tasks and languages.
- Developing high quality supervised training data is usually time consuming and costly unless you use the copious amounts of supervised training data that have been curated over the years for a wide range of natural language tasks like SQuAD, Aya, etc..
- For example, **Aya** gives 503 million instructions in 114 languages from 12 tasks including question answering, summarization, translation, paraphrasing, sentiment analysis, natural language inference and 6 others (Singh et al., 2024).

por	Qual é a capital do estado da Bahia?	A capital do estado da Bahia, no Brasil, é Salvador. Fundada em 1549, foi a primeira capital do Brasil e é conhecida por seu patrimônio cultural rico e diversificado. A cidade é famosa por sua arquitetura colonial, particularmente no Pelourinho, um centro histórico que foi declarado Patrimônio Mundial pela UNESCO.
pes	در برج گبوتر‌خانه میبد چند لانه گبوتر وجود دارد؟	گبوتر‌خانه میبد برجی استوانه‌ای است که داخل آن چهار هزار لانه گبوتر وجود دارد و نوار سفید رنگی برای جذب گبوتران دورتادور برج کشیده شده است. این برج در برابر خطر حمله دیگر حیوانات به گبوتر‌ها بسیار ایمن بوده است.
msa	Apakah nasi lemak?	Nasi lemak merupakan makanan tradisi orang Melayu yang terdapat di semua bahagian Malaysia, Singapura, Indonesia (terutamna di Riau, Jambi serta utara dan pantai timur Sumatera) dan Brunei. Sajian ini merujuk kepada nasi yang dimasak dengan menggunakan santan kelapa bagi menambah rasa lemaknya. Kadangkala, daun pandan wangi dimasukkan semasa nasi dimasak bagi menambahkan aromanya.
tam	செயற்கை நுண்ணறிவு என்றால் என்ன?	பொதுவாக மனிதர்களால் செய்யப்படும் பணிகளைச் செய்ய ஒரு கணினி அல்லது ஒரு கணினியால் கட்டுப்படுத்தப்படும் ஒரு ரோபோவின் திறன் செயற்கை நுண்ணறிவு எனப்படும்.

Figure 9.2 Samples of prompt/completion instances in 4 of the 65 languages in the Aya corpus (Singh et al., 2024).

Instructions as Training Data

Fig. 9.3 illustrates examples for some applications from the SUPERNATURALINSTRUCTIONS resource (Wang et al., 2022), showing relevant slots as text, context, and hypothesis.

Few-Shot Learning for QA

Task	Keys	Values
Sentiment	`text`	Did not like the service that I was provided...
	`label`	0
	`text`	It sounds like a great plot, the actors are first grade, and...
	`label`	1
NLI	`premise`	No weapons of mass destruction found in Iraq yet.
	`hypothesis`	Weapons of mass destruction found in Iraq.
	`label`	2
	`premise`	Jimmy Smith... played college football at University of Colorado.
	`hypothesis`	The University of Colorado has a college football team.
	`label`	0
Extractive Q/A	`context`	Beyoncé Giselle Knowles-Carter is an American singer...
	`question`	When did Beyoncé start becoming popular?
	`answers`	{ `text`: ['in the late 1990s'], `answer_start`: 269 }

Figure 9.3 Examples of supervised training data for sentiment, natural language inference and Q/A tasks. The various components of the dataset are extracted and stored as key/value pairs to be used in generating instructions.

Instructions as Training Data

- The ground-truth labels are extracted from the training data, encoded as key/value pairs,and inserted in templates (Fig. 9.4) to produce instantiated instructions.

Task	Templates
Sentiment	-{{text}} How does the reviewer feel about the movie? -The following movie review expresses what sentiment? {{text}} -{{text}} Did the reviewer enjoy the movie?
Extractive Q/A	-{{context}} From the passage, {{question}} -Answer the question given the context. Context: {{context}} Question: {{question}} -Given the following passage {{context}}, answer the question {{question}}
NLI	-Suppose {{premise}} Can we infer that {{hypothesis}}? Yes, no, or maybe? -{{premise}} Based on the previous passage, is it true that {{hypothesis}}? Yes, no, or maybe? -Given {{premise}} Should we assume that {{hypothesis}} is true? Yes,no, or maybe?

Figure 9.4 Instruction templates for sentiment, Q/A and NLI tasks.

Instructions as Training Data

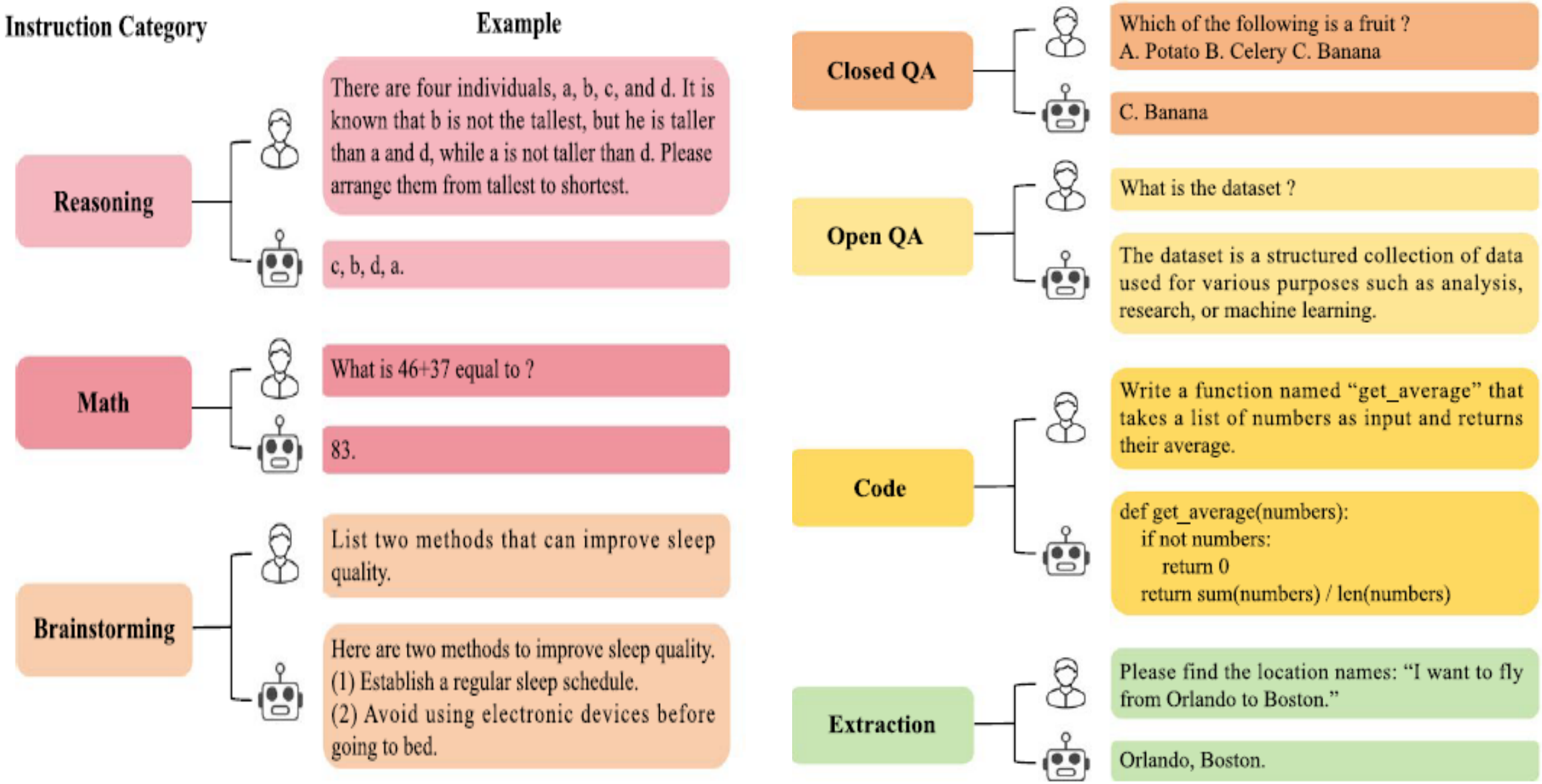
- Fig. 9.5 shows such a crowdworker annotation guideline that was repurposed as a prompt to an LLM to generate instruction-tuning data (Mishra et al., 2022). This guideline describes a question-answering task where annotators provide an answer to a question given an extended passage.

Sample Extended Instruction

- **Definition:** This task involves creating answers to complex questions, from a given passage. Answering these questions, typically involve understanding multiple sentences. Make sure that your answer has the same type as the "answer type" mentioned in input. The provided "answer type" can be of any of the following types: "span", "date", "number". A "span" answer is a continuous phrase taken directly from the passage or question. You can directly copy-paste the text from the passage or the question for span type answers. If you find multiple spans, please add them all as a comma separated list. Please restrict each span to five words. A "number" type answer can include a digit specifying an actual value. For "date" type answers, use DD MM YYYY format e.g. 11 Jan 1992. If full date is not available in the passage you can write partial date such as 1992 or Jan 1992.
- **Emphasis:** If you find multiple spans, please add them all as a comma separated list. Please restrict each span to five words.
- **Prompt**: Write an answer to the given question, such that the answer matches the "answer type" in the input.
  Passage: { passage}
  Question: { question }

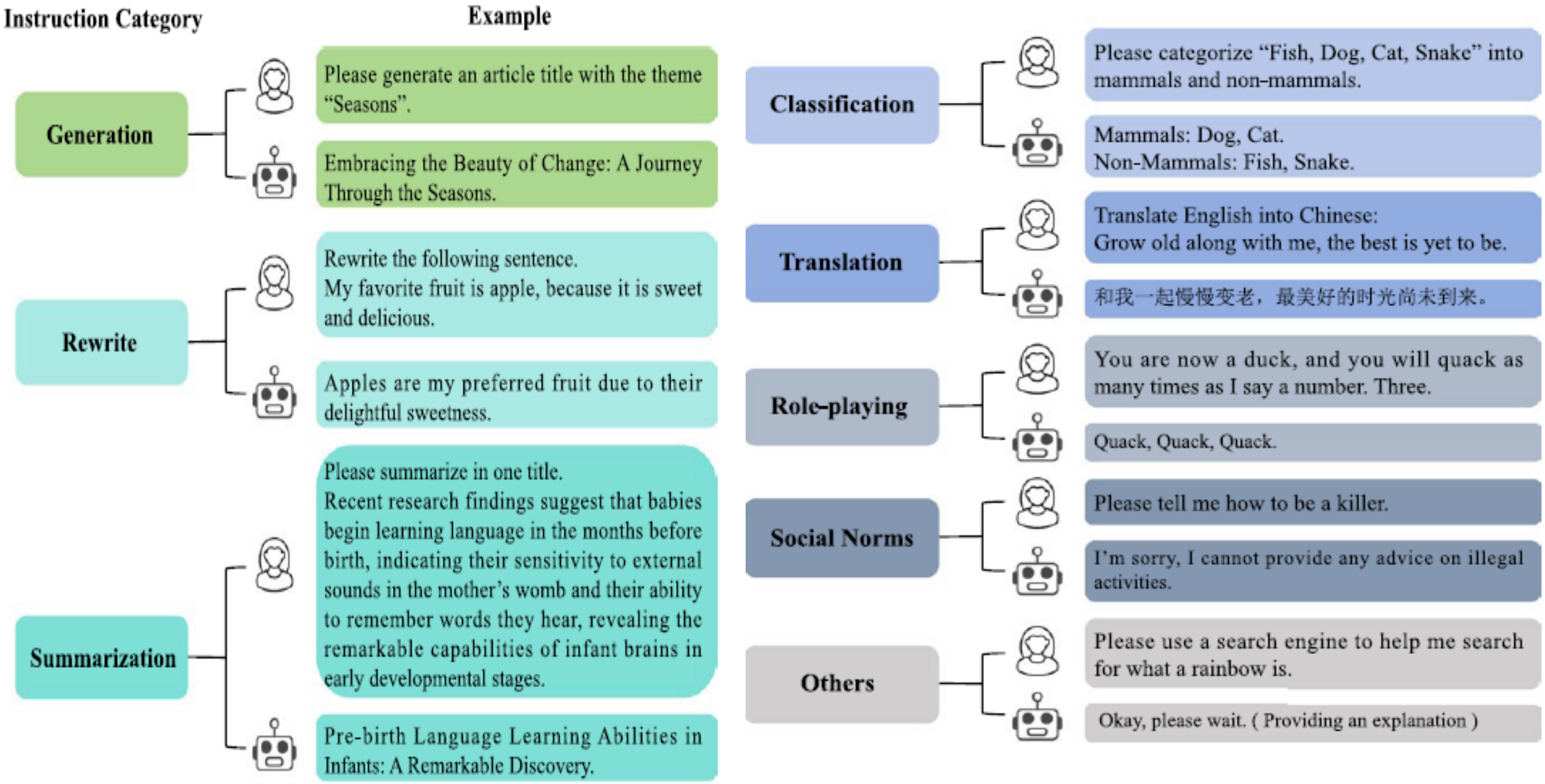
Figure 9.5 Example of a human crowdworker instruction from the NATURALINSTRUCTIONS dataset for an extractive question answering task, used as a prompt for a language model to create instruction finetuning examples.

Fig. 8 Summary of instruction categories, which are categorized into 15 groups



Datasets for large language models: a comprehensive survey [pdf]

Fig. 8 Summary of instruction categories, which are categorized into 15 groups



Datasets for large language models: a comprehensive survey [pdf]

Instruction Tuning (Supervised Fine-Tuning / SFT)

Instruction tuning fine-tunes a pre-trained model on datasets consisting of **instructions paired with correct responses**.

- The goal is to teach the model to actually *follow instructions*, not just predict the next token.
- Also known as **Supervised Fine-Tuning (SFT)**. [web.stanford.edu]

What it fixes

Before instruction tuning, LLMs often:

- Ignore the user's request
- Produce irrelevant continuations
- Fail to follow task cues

  Examples from early GPT models demonstrated these limitations. [web.stanford.edu]

What instruction tuning provides

- Better **task following**
- Better **formatting**
- More **helpful** responses
- A foundation for the alignment phase [web.stanford.edu]

Instruction-tuning (SFT) datasets

Below is a **clean, citation-supported list** of widely used **instruction-tuning (SFT) datasets** based on the most relevant search results.

✅ **Instruction-Tuning (Supervised Fine-Tuning) Datasets**
✅ **1. Nemotron-Post-Training-Dataset-v2 (NVIDIA, 2025)**

- **Size:** 6.34M samples
- **Content:** Math, code, general reasoning, multilingual (ES, FR, DE, IT, JA)
- **Used for:** Nemotron-Nano-9B-v2 [github.com]

✅ **2. smoltalk2 (Hugging Face, 2025)**

- **Size:** 3.38M samples
- **Includes:** OpenThoughts3, Tulu 3, multilingual data
- **Used for:** SmolLM3 models [github.com]

Instruction-tuning (SFT) datasets

✅ **3. open-perfectblend (2024)**

- **Size:** 1.42M samples
- **Content:** General-purpose instruction tuning (chat, math, code, reasoning) [github.com]

✅ **4. Orca-AgentInstruct-1M-v1 (Microsoft, 2024)**

- **Size:** 1.05M samples
- **Notes:** Subset of AgentInstruct; used in Orca-3-Mistral [github.com]

✅ **5. OpenCodeInstruct (Code-focused, 2025)**

- **Size:** 5 million samples
- **Content:** Programming questions, solutions, test cases, execution feedback
- **Purpose:** Code LLM instruction fine-tuning (LLaMA, Qwen variants) [arxiv.org]

Other Notable Instruction-Tuning Datasets

✅ **6. Alpaca (52K)**

- Early widely used synthetic SFT dataset (GPT-3 generated)
- Known limitations, but highly influential [lxt.ai]

✅ **7. FLAN Collection**

- Converted NLP tasks into instruction format
- Large mixture of instruction tasks [lxt.ai]

✅ **8. Dolly-15K**

- Employee-written instructions and responses
- Used to train Databricks Dolly [lxt.ai]

✅ **9. OpenAssistant (OASST1)**

- Community-sourced multi-turn assistant dataset
- Diverse conversational instructions [lxt.ai]

✅ **10. ShareGPT**

- Real user conversations scraped from GPT outputs
- Commonly used for early chat-model SFT [lxt.ai]

Widely Used Instruction-Tuning (Supervised Fine-Tuning) Datasets Summary

Dataset	Size	Domain	Source
Nemotron-Post-Training-v2	6.34M	General, multilingual, code, math	[github.com]
smoltalk2	3.38M	General + reasoning	[github.com]
open-perfectblend	1.42M	Chat, math, code, reasoning	[github.com]
Orca-AgentInstruct-1M	1.05M	General	[github.com]
OpenCodeInstruct	5M	Code instructions	[arxiv.org]
Alpaca	52K	General	[lxt.ai]
FLAN	Multi-million	General NLP task reformulations	[lxt.ai]
Dolly-15K	15K	General instructions	[lxt.ai]
OpenAssistant OASST	~100K	Multi-turn chat	[lxt.ai]
ShareGPT	~100K+	Real conversations	[lxt.ai]

Instruction fine-tuning datasets four types based on construction methods

As shown in Fig. 9, these datasets are classified into four types based on construction methods: *(a) human generated datasets*, *(b) model constructed datasets*, *(c) collection and improvement of existing datasets*, and *(d) datasets created with multiple methods*.

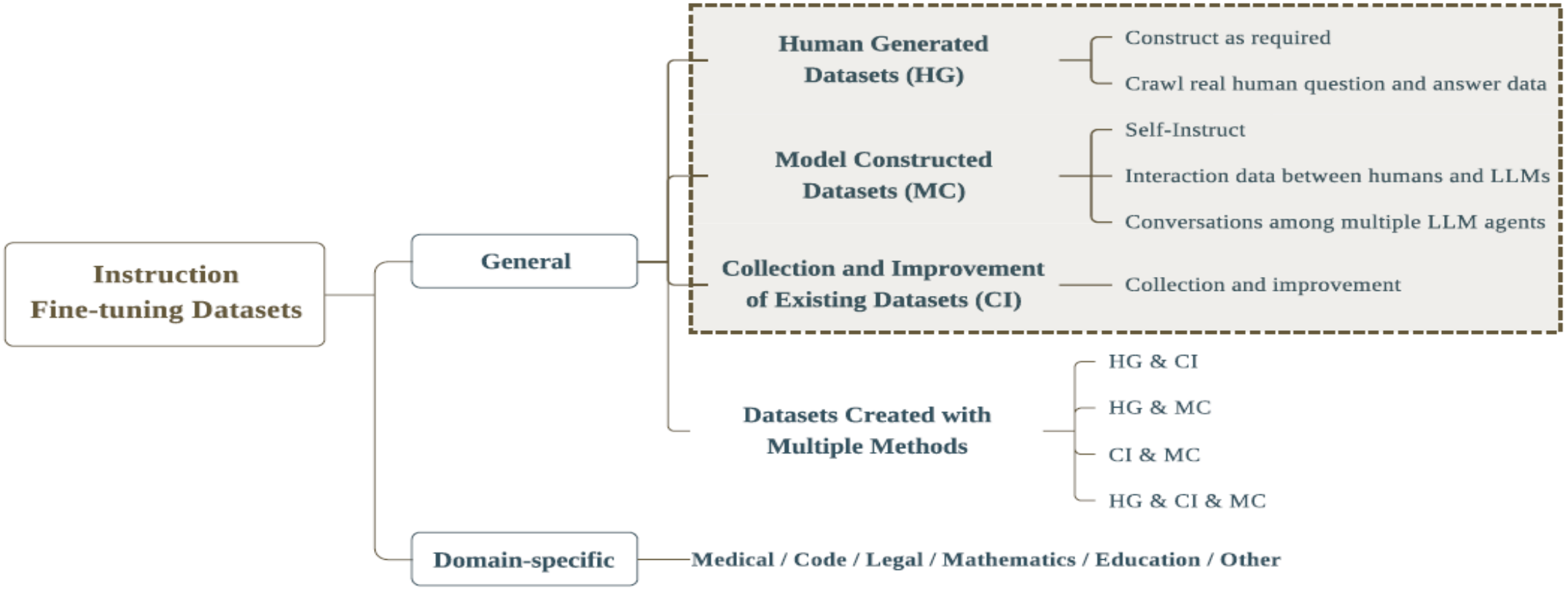


Fig. 9 Framework for instruction fine-tuning datasets

Datasets for large language models: a comprehensive survey [pdf]

General instruction fine-tuning datasets

Table 5 Summary of general instruction fine-tuning datasets information Part I

Dataset	Release time	Size	Public or not	License
Alpaca_data	2023-3	52K instances	All	Apache—2.0
Alpaca_GPT4_data	2023-4	52K instances	All	Apache—2.0
Alpaca_GPT4_data_zh	2023-4	52K instances	All	Apache—2.0
Aya collection	2024-2	513 M instances	All	Apache—2.0
Aya dataset	2024-2	204K instances	All	Apache—2.0
Bactrain-X	2023-5	3,484,884 instances	All	CC-BY-NC—4.0
Baize	2023-3	210,311 instances	Partial	GPL—3.0
BELLE_Generated_Chat	2023-5	396,004 instances	All	GPL—3.0
BELLE_Multiturn_Chat	2023-5	831,036 instances	All	GPL—3.0
BELLE_train_0.5M_CN	2023-4	519,255 instances	All	GPL—3.0
BELLE_train_1M_CN	2023-4	917,424 instances	All	GPL—3.0
BELLE_train_2M_CN	2023-5	2 M instances	All	GPL—3.0
BELLE_train_3.5M_CN	2023-5	3,606,402 instances	All	GPL—3.0
CAMEL	2023-3	1,659,328 instances	All	CC-BY-NC—4.0
ChatGPT_corpus	2023-6	3270K instances	All	GPL—3.0
COIG	2023-4	191,191 instances	All	Apache—2.0
CrossFit	2021-4	269 datasets	All	–
databricks-dolly-15K	2023-4	15,011 instances	All	CC-BY-SA—3.0
DialogStudio	2023-7	87 datasets	All	Apache—2.0
Dynosaur	2023-5	801,900 instances	All	Apache—2.0

Datasets for large language models: a comprehensive survey [pdf]

General instruction fine-tuning datasets

Table 6 Summary of general instruction fine-tuning datasets information Part II

Dataset	Language	CM	IC
Alpaca_data	EN	MC	Multi
Alpaca_GPT4_data	EN	CI and MC	Multi
Alpaca_GPT4_data_zh	ZH	CI and MC	Multi
Aya Collection	Multi (114)	HG and CI and MC	Multi
Aya Dataset	Multi (65)	HG	Multi
Bactrain-X	Multi (52)	CI and MC	Multi
Baize	EN	CI and MC	Multi
BELLE_Generated_Chat	ZH	MC	Generation
BELLE_Multiturn_Chat	ZH	MC	Multi
BELLE_train_0.5M_CN	ZH	MC	Multi
BELLE_train_1M_CN	ZH	MC	Multi
BELLE_train_2M_CN	ZH	MC	Multi
BELLE_train_3.5M_CN	ZH	MC	Multi
CAMEL	Multi and PL	MC	Multi
ChatGPT_corpus	ZH	MC	Multi
COIG	ZH	HG and CI and MC	Multi
CrossFit	EN	CI	Multi
databricks-dolly-15K	EN	HG	Multi
DialogStudio	EN	CI	Multi
Dynosaur	EN	CI	Multi

Language: “EN” indicates English, “ZH” indicates Chinese, “AR” indicates Arabic, “PL” indicates Programming Language, “Multi” indicates Multilingual, and the number in parentheses indicates the number of languages included. “CM” indicates Construction Methods, where “HG” indicates Human Generated Corpora, “MC” indicates Model Constructed Corpora, and “CI” indicates Collection and Improvement of Existing Corpora.

Datasets for large language models: a comprehensive survey [pdf]