模式分析与机器智能
工业和信息化部重点实验室
MIIT Key Laboratory of
Pattern Analysis & Machine Intelligence

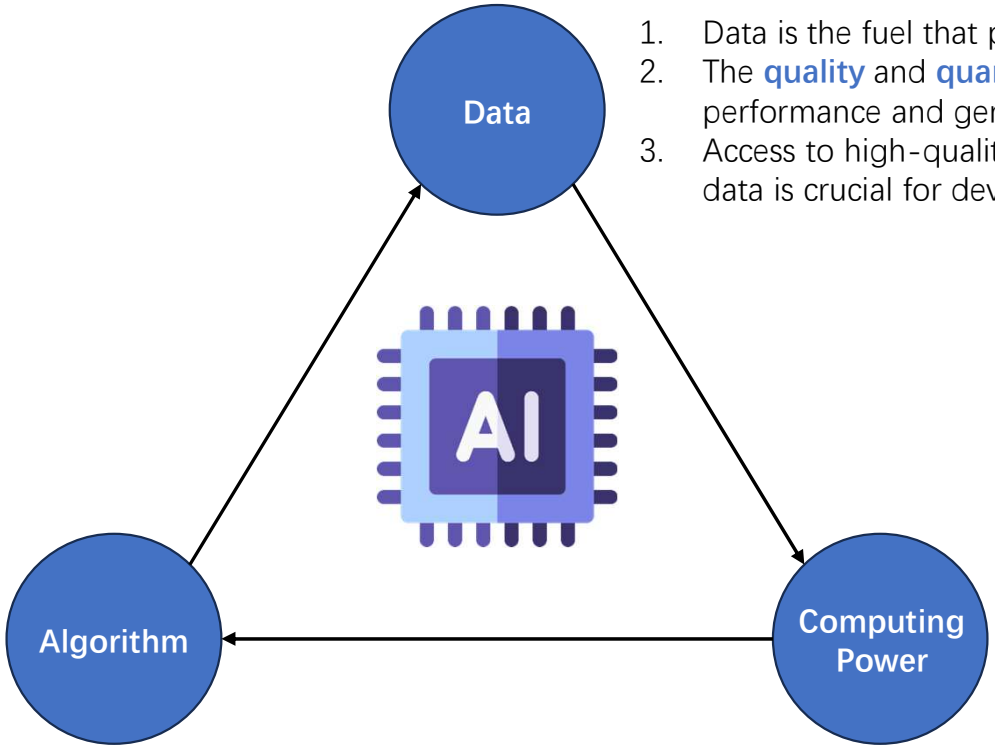
ParNeC 模式识别与神经计算研究组
PAttern Recognition and NEural Computing

Code LLMs and Data Selection

Weijie Lv

2024.4.22

Introduction

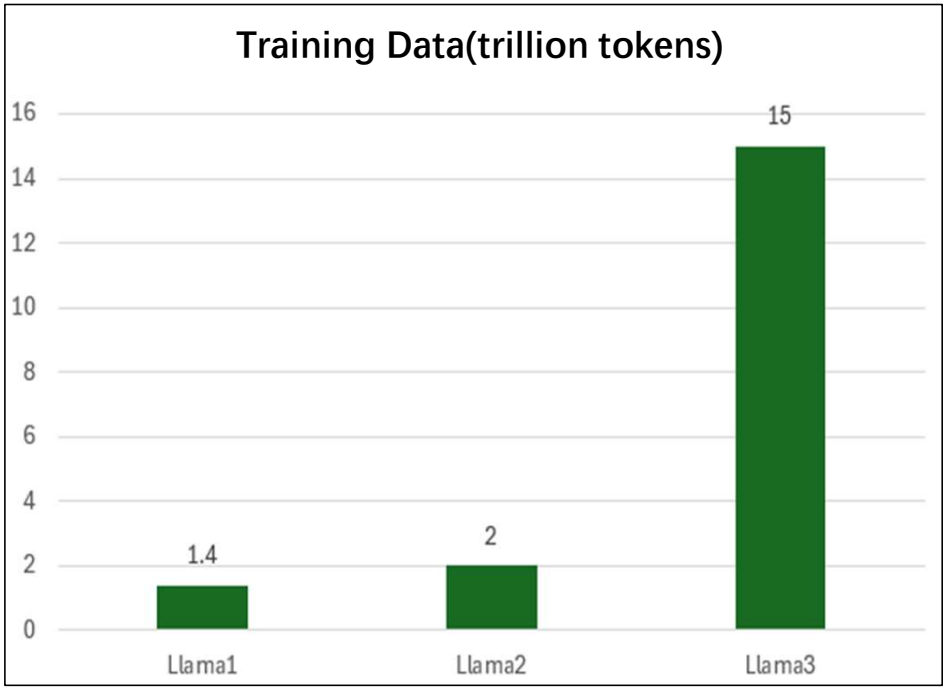


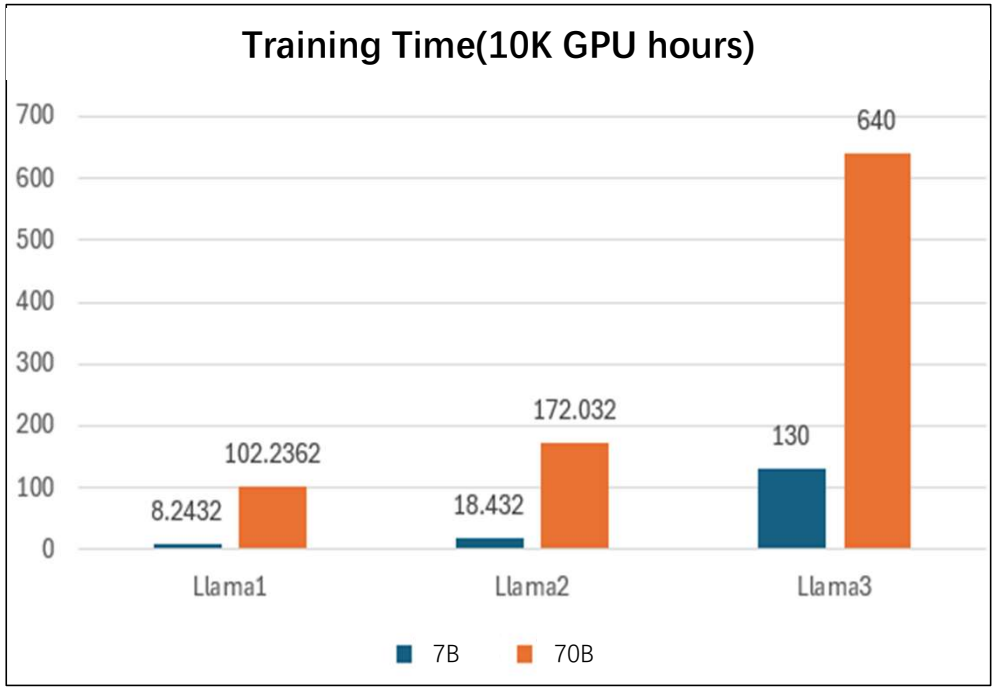
1. Data is the fuel that powers AI models and algorithms.
2. The **quality** and **quantity** of data significantly impact the performance and generalization capabilities of AI models.
3. Access to high-quality, well-labeled, and representative data is crucial for developing successful AI applications.

1. Algorithms are the mathematical and computational models that drive AI systems.
2. Deep learning algorithms (e.g., **CNN** and **Transformer**) have revolutionized various domains, including computer vision, natural language processing, and speech recognition.

1. AI algorithms, especially deep learning models, require immense computational resources for training and inference.
2. Powerful hardware, such as **GPUs** (Graphics Processing Units) and **TPUs** (Tensor Processing Units), accelerate the training and deployment of AI models.

2024.4.19 Llama 3





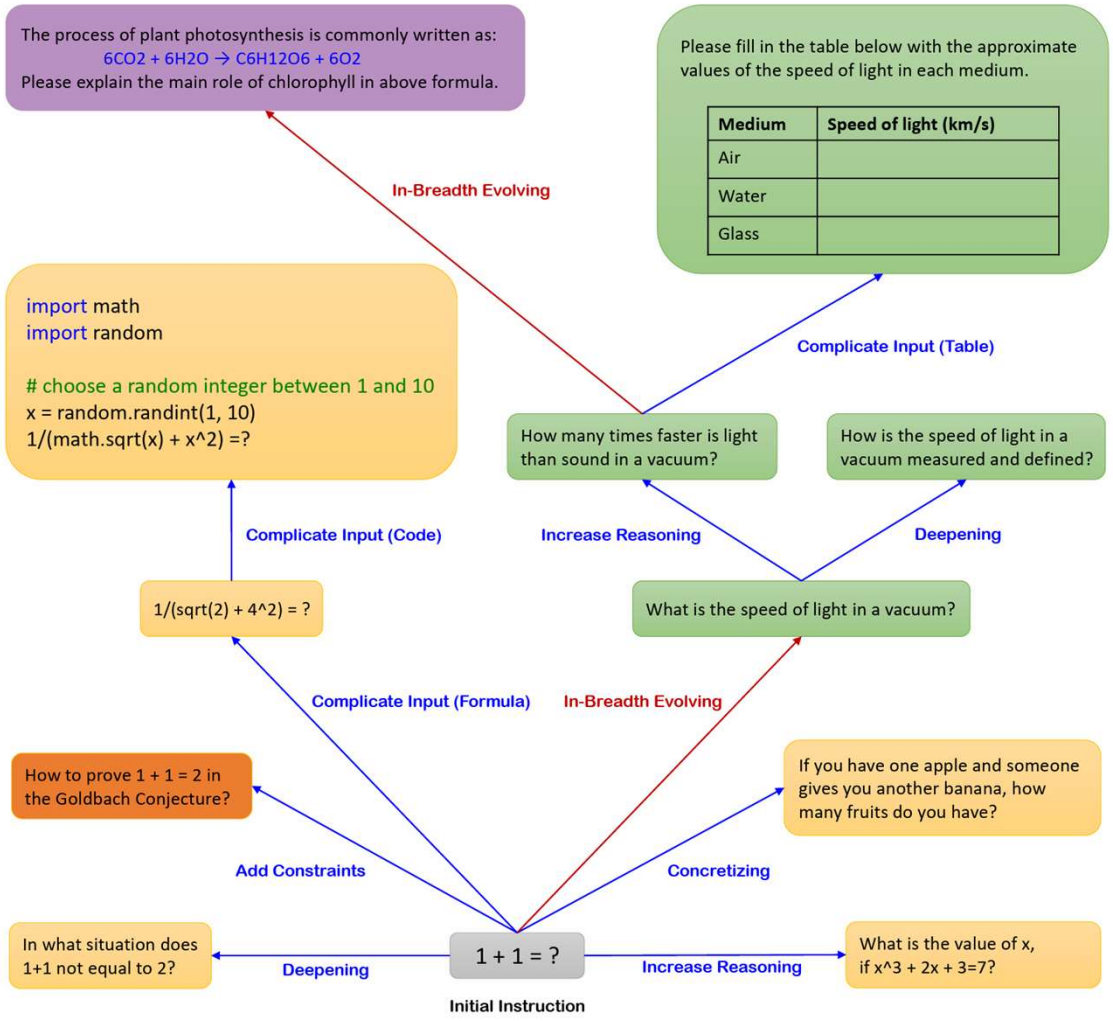
Taking the 70 billion parameter scale Llama3-70B as an example, the training time is 6.4 million GPU hours. Using AWS's p4d.24xlarge example, which contains 8 A100s, the pay-as-you-go **8 card is $32.77 per hour**, 6.4 million GPU hours is **800,000 such machines**, and the pay-as-you-go price is **$800,000 × $32.77 = $26,216,000**.

Unbearable for individual developers!!!

2306 WizardCoder (ICLR2024)

Evol-Instruct leverages the capabilities of Large Language Models (LLMs) to automatically generate open-domain instructions of varying difficulty levels.

- **In-Depth Evolving**: This process enhances instructions by ***making them more complex*** through various techniques such as *adding constraints*, *deepening the inquiry*, *concretizing concepts*, *increasing reasoning steps*, and *complicating input*.
- **In-Breadth Evolving**: This process aims to ***increase the diversity*** and coverage of topics and skills by creating new instructions that are more *rare* or *specific*, thus expanding the range of tasks the model can handle.



2306 WizardCoder (ICLR2024)

define roles ⇢ I want you act as a Prompt Creator.
Your goal is to draw inspiration from the #Given Prompt# to create a brand
define goals ⇢ new prompt. This new prompt should belong to the same domain as the #Given Prompt# but be even more rare.
The LENGTH and complexity of the #Created Prompt# should be similar to that of the #Given Prompt#.
define rules ⇢ The #Created Prompt# must be reasonable and must be understood and responded by humans.
'#Given Prompt#', '#Created Prompt#', 'given prompt' and 'created prompt' are not allowed to appear in #Created Prompt#
#Given Prompt#:
Create a JavaScript function to insert an item at a specified index in an array.
#Created Prompt#:

①

GPT-3.5

②

③

Develop a JavaScript function to replace all occurrences of a specified element in an array with a new element.

④

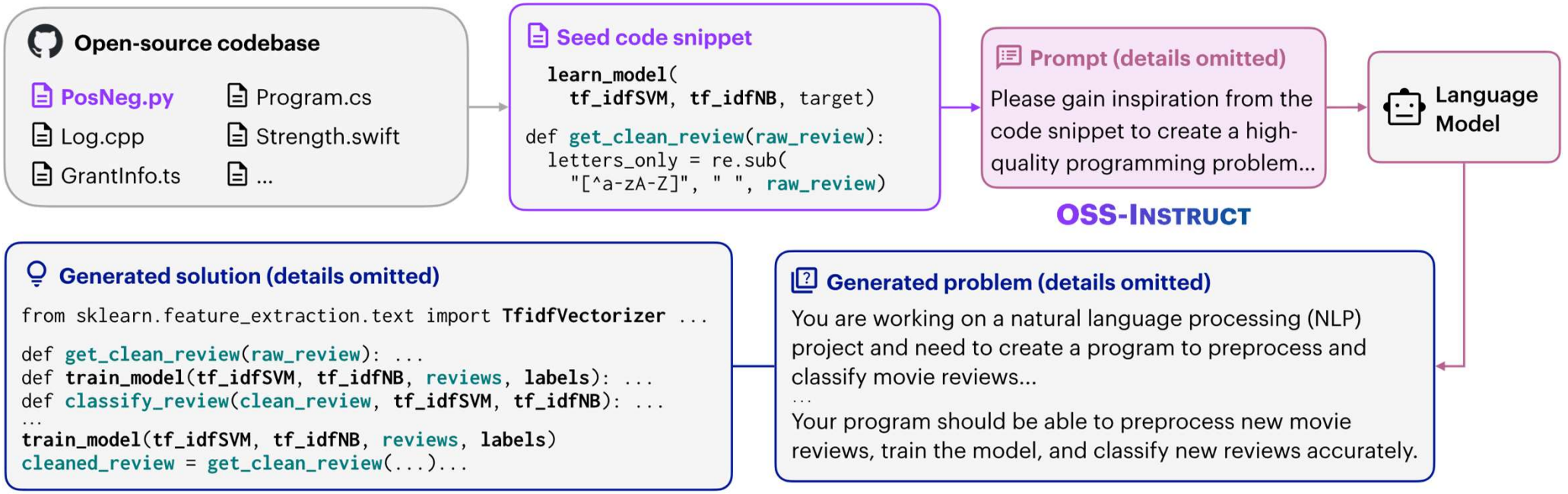
```
function replaceElements(array,
oldElement, newElement) {
  return array.map(item =>
item === oldElement ?
newElement : item);
}
```

2306 WizardCoder (ICLR2024)

Table 1: Results of pass@1(%) on HumanEval and MBPP. Most scores are retrieved from the papers of StarCoder [11] and CodeT5+ [18]. We follow the previous works [31] to generate n samples to estimate the pass@1 score with the same set of hyper-parameters: temperate=0.2, and top_p=0.95. *: we evaluate this model by ourselves.

Model	Params	HumanEval	MBPP
Closed-source models			
LaMDA [40]	137B	14.0	-
AlphaCode [12]	1.1B	17.1	-
PaLM [3]	540B	26.2	36.8
PaLM-Coder [3]	540B	36.0	47.0
PaLM 2-S [4]	-	37.6	50.0
Codex [16]	2.5B	21.4	-
Codex [16]	12B	28.8	-
Code-Cushman-001 [38]	-	33.5	45.9
Code-Davinci-002 [38]	-	47.0	58.1
GPT-3.5 [2]	-	48.1	-
GPT-4 [2]	-	67.0	-
Open-source models			
LLaMa [8]	33B	21.7	30.2
LLaMa [8]	65B	23.7	37.7
CodeGen-Multi [13]	16B	18.3	20.9
CodeGen-Mono [13]	16B	29.3	35.3
CodeGeeX [14]	13B	22.9	24.4
StarCoder [11]	15B	33.6	43.6*
CodeT5+ [18]	16B	30.9	-
InstructCodeT5+ [18]	16B	35.0	-
WizardCoder	15B	**57.3** (+22.3)	**51.8** (+8.2)

2312 Magicoder



OSS-Instruct aims to improve the performance of Large Language Models (LLMs) in code generation tasks by leveraging open-source code snippets. *This method is designed to address the inherent bias present in synthetic data generated by LLMs by providing them with a wealth of open-source references*, enabling the production of more diverse, realistic, and controllable data for instruction tuning.

2312 Magicoder

define roles ⇢

define goals ⇢

define rules ⇢

```
You are exceptionally skilled at crafting high-quality programming problems and
offering precise solutions.

Please gain inspiration from the following random code snippet to create a
high-quality programming problem. Present your output in two distinct sections:
[Problem Description] and [Solution].

Code snippet for inspiration:
```
{code}
```

Guidelines for each section:
 1. [Problem Description]: This should be **completely self-contained**, providing
    all the contextual information one needs to understand and solve the problem.
    Assume common programming knowledge, but ensure that any specific context,
    variables, or code snippets pertinent to this problem are explicitly included.
 2. [Solution]: Offer a comprehensive, **correct** solution that accurately
    addresses the [Problem Description] you provided.
```

Figure 2: The detailed prompt design for OSS-INSTRUCT

2312 Magicoder

Seed: library imports

```
import numpy as np
import gym_electric_motor as gem
import matplotlib.pyplot as plt
```

Problem

Create a **reinforcement learning agent** to control an electric motor using the OpenAI **Gym** environment...

Code

```
import numpy as np
import gym_electric_motor as gem
...
env = gem.make("DcSeriesCont-v1")
class DQNAgent:
  def __init__(self, state_dim, action_dim): ...
  def build_model(self): ...
  def act(self, state): ...
  def train(self, state, action, reward, ...): ...
...
for episode in range(episodes):
    state = env.reset()
    state = np.reshape(state, [1, state_dim])
    ...
```

Seed: comments

```
# Set degrees
```

Problem

Implement a Python class that represents a **temperature in degrees**...

Code

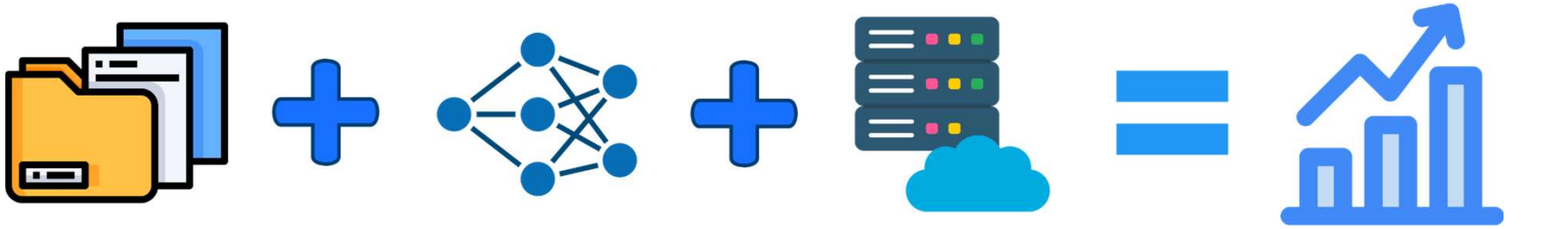
```
class TemperatureConverter:
  def __init__(self): ...
  def set_celsius(self, degrees): ...
  def set_fahrenheit(self, degrees): ...
  def set_kelvin(self, degrees): ...
  def get_celsius(self): ...
  def get_fahrenheit(self): ...
  def get_kelvin(self): ...
  def convert_to(self, unit):
      if unit == 'C':
          return self.get_celsius()
      elif unit == 'F':
          return self.get_fahrenheit()
      elif unit == 'K':
          return self.get_kelvin()
      ...
```

2312 Magicoder

Table 1: Pass@1 (%) results of different LLMs on HumanEval (+) and MBPP (+) computed with greedy decoding. The abbreviations "CL" and "SC" refer to the base models CODELLAMA-PYTHON and StarCoder, respectively. We report the results consistently from the EvalPlus [Liu et al., 2023b] Leaderboard.

Model	Release Date	Size	Benchmark		Open-Source	
			HumanEval (+)	MBPP (+)	Weight	Data
GPT-3.5 Turbo	Nov 2023	-	72.6 (65.9)	81.7 (69.4)	○	○
GPT-4 Turbo	Nov 2023	-	**85.4 (81.7)**	**83.0 (70.7)**	○	○
CODELLAMA-PYTHON	Aug 2023	34B	51.8 (42.7)	67.2 (52.9)	●	○
WizardCoder-CL	Sep 2023	34B	73.2 (64.6)	73.2 (59.9)	●	○
CodeT5+	May 2023	16B	31.7 (26.2)	54.6 (44.4)	●	●
CodeGen-Mono	Mar 2022	16B	32.9 (27.4)	52.6 (43.6)	●	●
StarCoder	May 2023	15B	34.1 (29.3)	55.1 (46.1)	●	●
CODELLAMA-PYTHON	Aug 2023	13B	42.7 (36.6)	61.2 (50.9)	●	○
WizardCoder-SC	Sep 2023	15B	51.9 (45.1)	61.9 (50.6)	●	○
StarCoder	May 2023	7B	24.4 (20.7)	33.1 (28.8)	●	●
Mistral	Oct 2023	7B	28.7 (23.2)	50.1 (40.9)	●	○
CodeT5+	May 2023	6B	29.3 (23.8)	51.9 (40.9)	●	●
CodeGen-Mono	Mar 2022	6B	29.3 (25.6)	49.9 (42.1)	●	●
CODELLAMA-PYTHON	Aug 2023	7B	37.8 (34.1)	57.6 (45.4)	●	○
WizardCoder-CL	Sep 2023	7B	48.2 (40.9)	56.6 (47.1)	●	○
Magicoder-CL	Dec 2023	7B	60.4 (55.5)	64.2 (52.6)	●	●
Magicoder$\mathcal{S}$-CL	Dec 2023	7B	**70.7 (66.5)**	**68.4 (56.6)**	●	●

Stage Summary



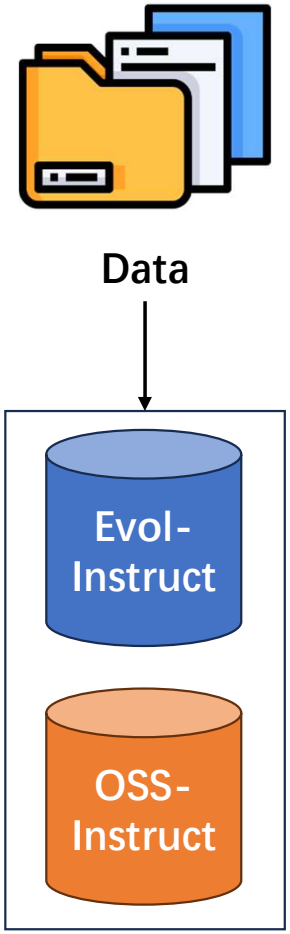
Data + Algorithm + Computing Power = High Performance

- Data
  - Evol-Instruct
  - OSS-Instruct
- Algorithm
  - CodeLlama
  - StarCoder
  - CodeGemma
  - DeepSeek Coder
- Computing Power

Full-Parameter Finetuning

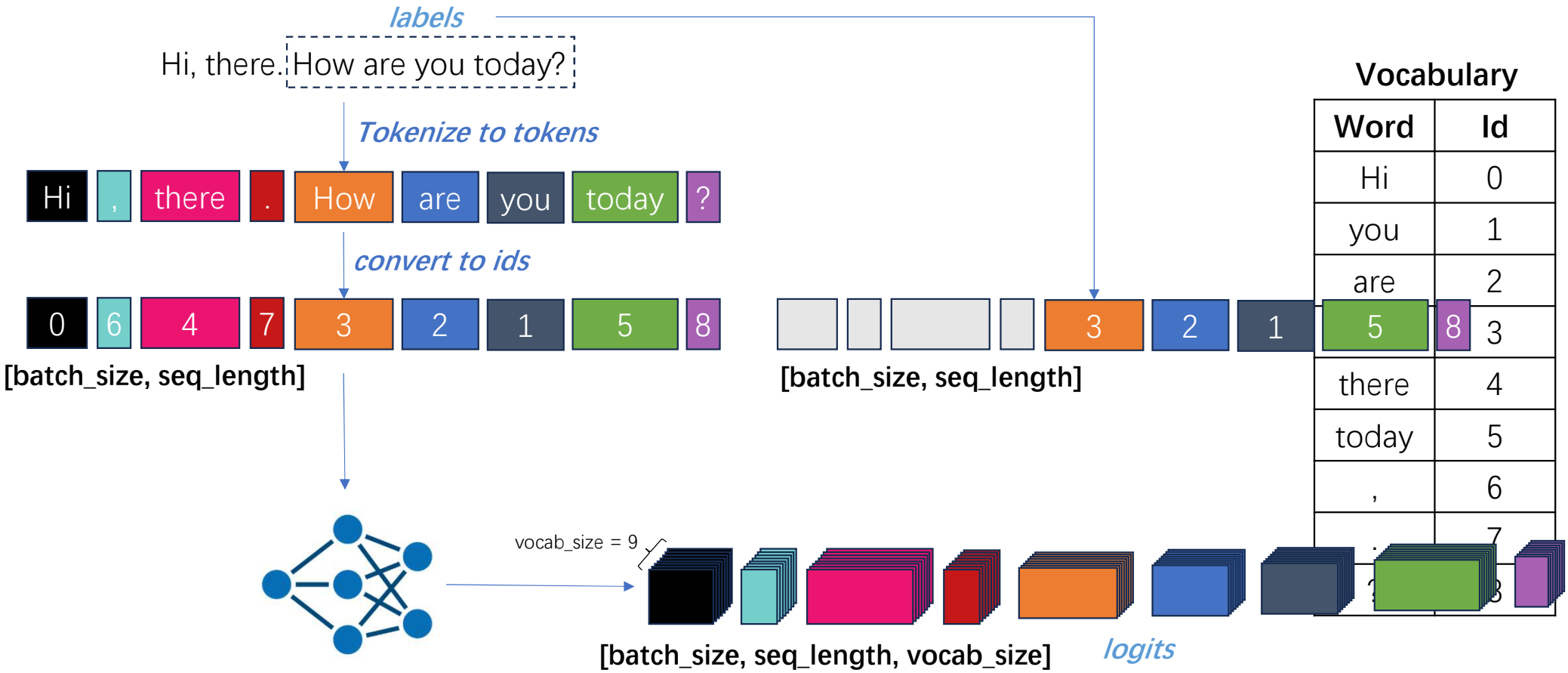
Model Size(7B, float16):
7 x 2 = 14GB
Gradient:
7 x 2 = 14GB
Optimizer Status(Adam):
2 x 7 x 2 = 28GB
Activation:
1 x 2 = 2GB
Total: at least *58GB !!!*

Stage Summary



Data	
Quantity	Quality
✓	?
Synthetic data from powerful LLMs can improve model's performance and reduce costs.	LIMA(Zhou et al., 2023) highlights that even a limited amount of manually curated, **high-quality data** can elevate the model's performance.

Next Token Prediction

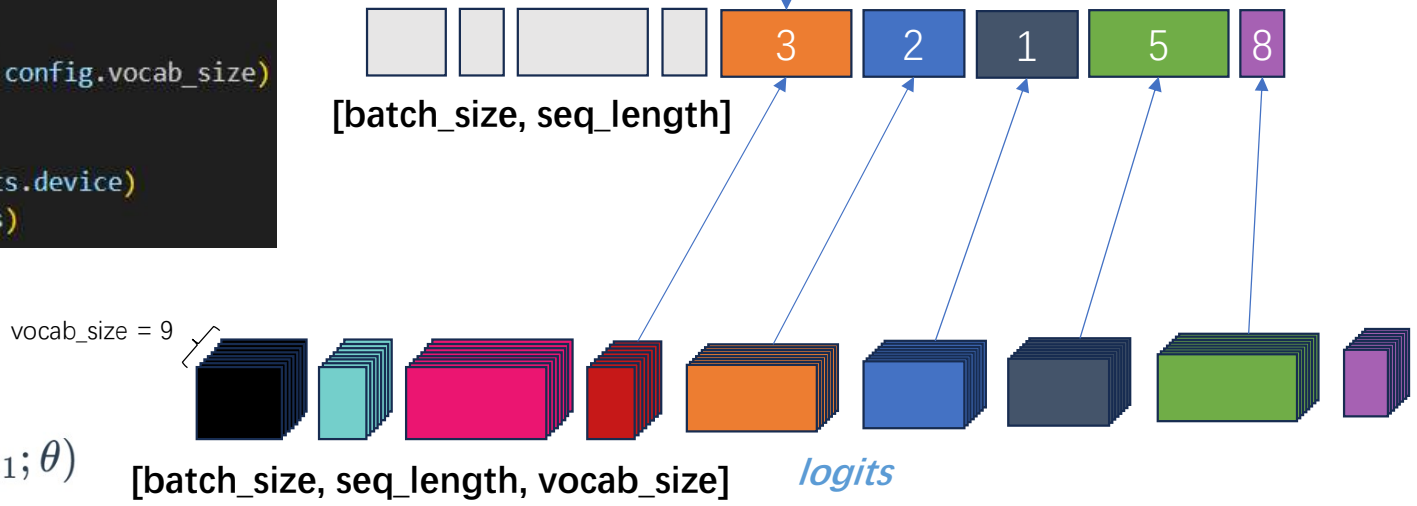


Next Token Prediction

labels

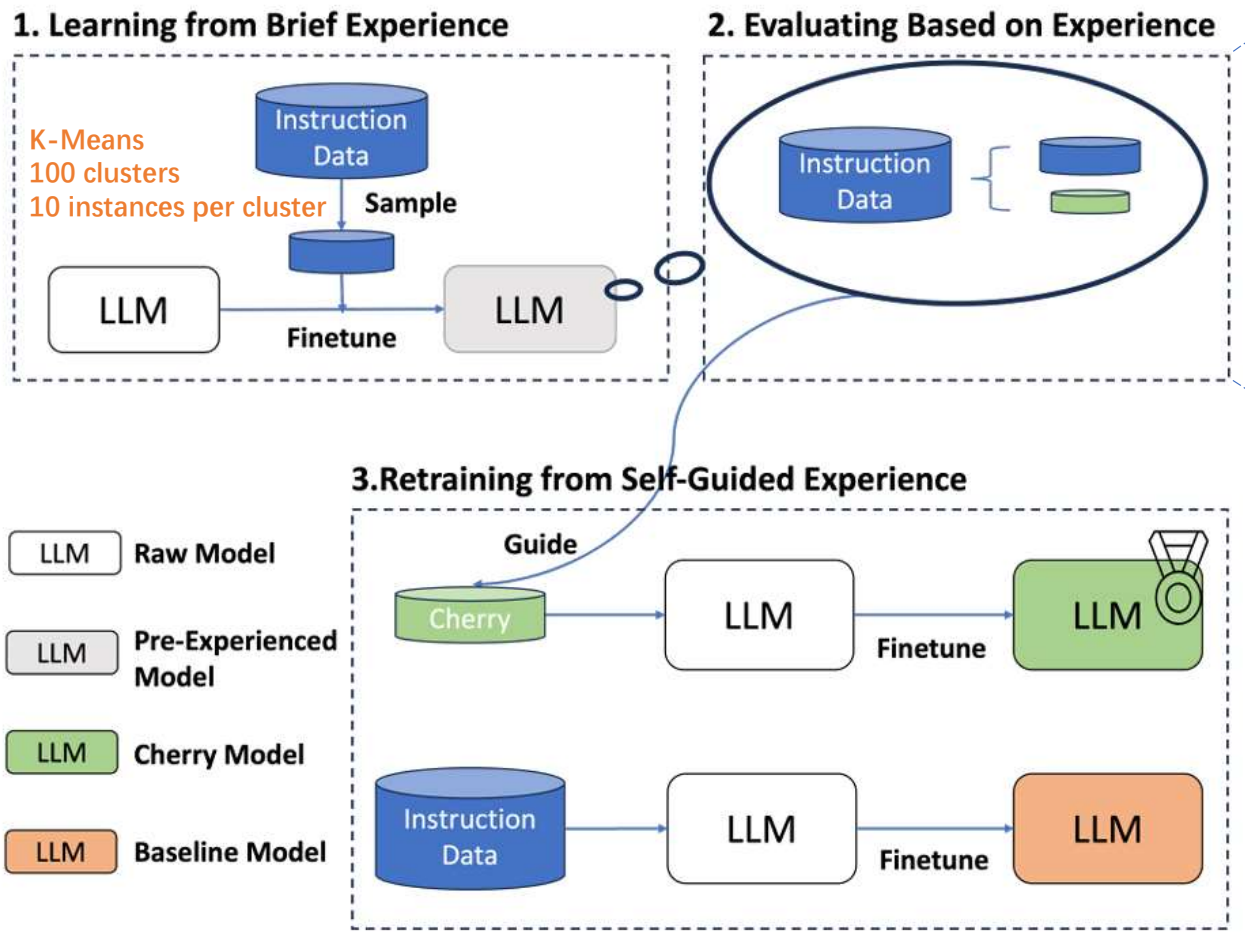
Hi, there. How are you today?

```python
loss = None
if labels is not None:
    # Shift so that tokens < n predict n
    shift_logits = logits[..., :-1, :].contiguous()
    shift_labels = labels[..., 1:].contiguous()
    # Flatten the tokens
    loss_fct = CrossEntropyLoss()
    shift_logits = shift_logits.view(-1, self.config.vocab_size)
    shift_labels = shift_labels.view(-1)
    # Enable model parallelism
    shift_labels = shift_labels.to(shift_logits.device)
    loss = loss_fct(shift_logits, shift_labels)
```



$$L_{\text{ntp}}(T) = -\frac{1}{N}\sum_{n=1}^{N}\log p(t_n|t_1,...,t_{n-1};\theta)$$

2308 Cherry LLM (NAACL 2024)



Instruction-Following Difficulty(IFD)

$$L_\theta(A|Q) = -\frac{1}{N}\sum_{i=1}^{N}\log P(w_i^A|Q, w_1^A, w_2^A, \ldots, w_{i-1}^A; \theta)$$

$$s_\theta(A) = -\frac{1}{N}\sum_{i=1}^{N}\log P(w_i^A|w_1^A, \ldots, w_{i-1}^A; \theta).$$

$$\mathrm{IFD}_\theta(Q, A) = \frac{s_\theta(A|Q)}{s_\theta(A)}$$

The score measures the degree how a given instruction benefits the alignment of the corresponding response. ***High IFD scores infer the in ability of the model to align responses to the given corresponding instructions, which in turn indicates the difficulty of an instruction.***

IFD: Positive Examples

[Instruction] Create a poem that tells the story of a struggle against an unseen force.
[Input]
[Response]
A distant enemy looms on the horizon,
Selves of the light fight despite fear and fatigue.
Innumerable obstacles barring their mission,
Though seeming to weaken, strengths within achieve.

Roots of bravery sustained their cause,
Though many obstacles pushed and pulled.
Heart's noise drowned out every pause,
Escaping each struggle stronger and full.

The reward of their labors thought clear and light,
The force of the shadows still kept them at bay.
Yet further pushed and tired through the night,
Triumph steadily came in a growing display.

Peace of their struggle found in the morning's glow,
Rising higher than any unseen force's ebb and flow.
[Direct Answer score (DA)]3.970
[Conditioned Answer score (CA)]3.337
[IFD score]0.928

[Instruction] Give a brief description of the coronavirus.
[Input]
[Response]
Coronavirus is a type of virus that can cause illnesses ranging from the common cold to severe diseases such as Middle East Respiratory Syndrome (MERS) and Severe Acute Respiratory Syndrome (SARS). The virus is named for the crown-like spikes on its surface. It spreads primarily through close contact with an infected person, through small droplets from the nose or mouth after coughing or sneezing. Symptoms can include fever, tiredness, and dry cough. In severe cases, it can cause difficulty breathing, pneumonia, kidney failure, and even death.
[Direct Answer score (DA)]0.761
[Conditioned Answer score (CA)]0.696
[IFD score]0.914

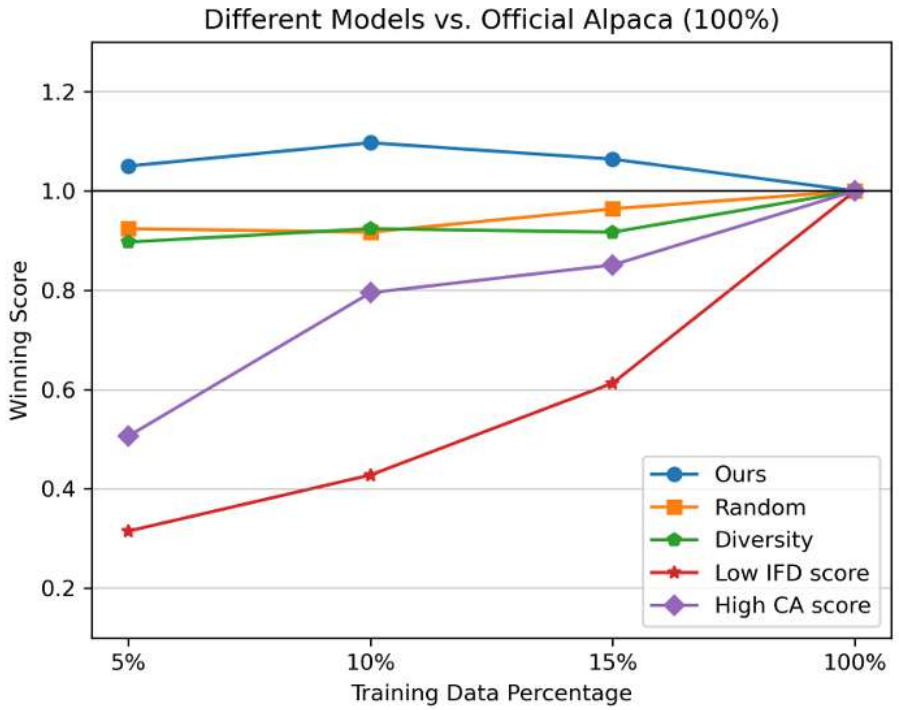
IFD: Negative Examples

[Instruction] Generate a report summarizing a conversation about car maintenance.
[Input] Joe and Lisa were discussing the importance of regular car maintenance such as oil changes, tire rotations, and brake inspects.
Joe believes that regular car maintenance is essential for keeping the car operating properly and avoiding major repairs. Lisa believes that some car maintenance procedures are unnecessary and are a waste of money.
[Response]
Joe and Lisa discussed the importance of regular car maintenance such as oil changes, tire rotations, and brake inspections. Joe believes that regular car maintenance is essential for keeping the car operating properly and avoiding major repairs while Lisa believes that some procedures are unnecessary and are a waste of money. It is agreed that regular oil changes and tire rotations are important to maintain the health of the car but that some other maintenance procedures can be skipped depending on the age and condition of the car.
[Direct Answer score (DA)] 1.667
[Conditioned Answer score] 0.599
[IFD score] 0.359

[Instruction] What emotion is expressed in this tweet?
[Input] I'm so frustrated right now!!
[Response] Frustration
[Direct Answer score (DA)] 6.593
[Conditioned Answer score (CA)] 0.601
[IFD score] 0.091

[Instruction] Identify a famous quote from the given book
[Input] Book: Pride and Prejudice by Jane Austen
[Response] "It is a truth universally acknowledged, that a single man in possession of a good fortune, must be in want of a wife."
[Direct Answer score (DA)] 0.497
[Conditioned Answer score] 0.026
[IFD score] 0.053

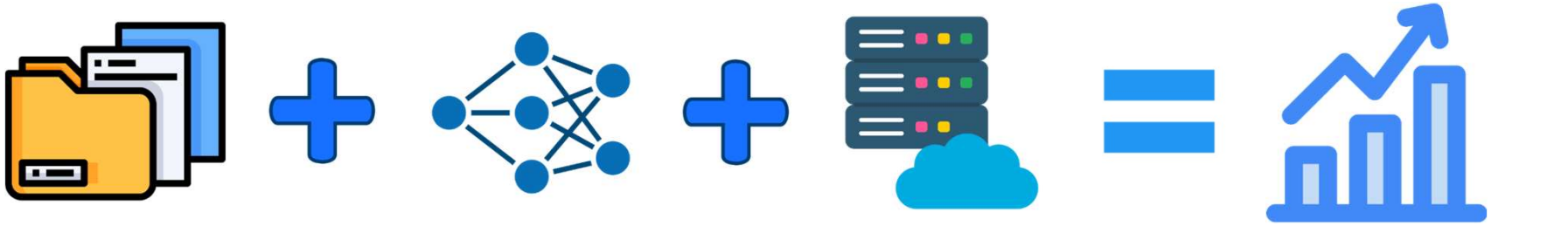
2308 Cherry LLM (NAACL 2024)



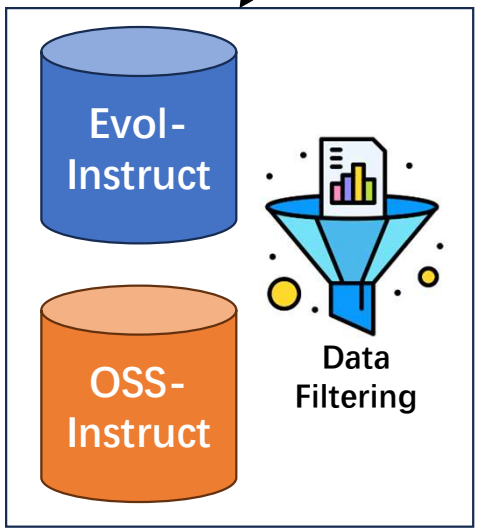
	Huggingface Open LLM Leaderboard					AlpacaEval
	Average	ARC	HellaSwag	MMLU	TruthfulQA	AlpacaEval
Official Alpaca	50.21	42.65	76.91	41.73	39.55	26.46
Ours (5% Alpaca)	52.06	53.92	79.49	36.51	38.33	34.74
Reimplemented WizardLM*	52.79	53.07	77.44	37.75	42.90	61.99
Ours (10% WizardLM)	51.59	52.90	78.95	33.08	41.41	61.44

Table 1: The comparison of performance on Huggingface Open LLM Leaderboard and AlpacaEval Leaderboard.

Summary



Data + Algorithm + Computing Power = Higher Performance



Full-Parameter Finetuning

Model Size(7B, float16):
7 x 2 = 14GB
Gradient:
7 x 2 = 14GB
Optimizer Status(Adam):
2 x 7 x 2 = 28GB
Activation:
1 x 2 = 2GB
Total: at least *58GB !!!*

CodeAcT: an efficient framework for data selection and fine-tuning for Code LLMs

Weijie Lv Hengbo Fan Xuan Xia

2024.4.22

CodeAcT

We present an efficient training framework tailored for large-scale code models. Our framework incorporates three key innovations to improve training efficiency and model performance while reducing computational requirements.

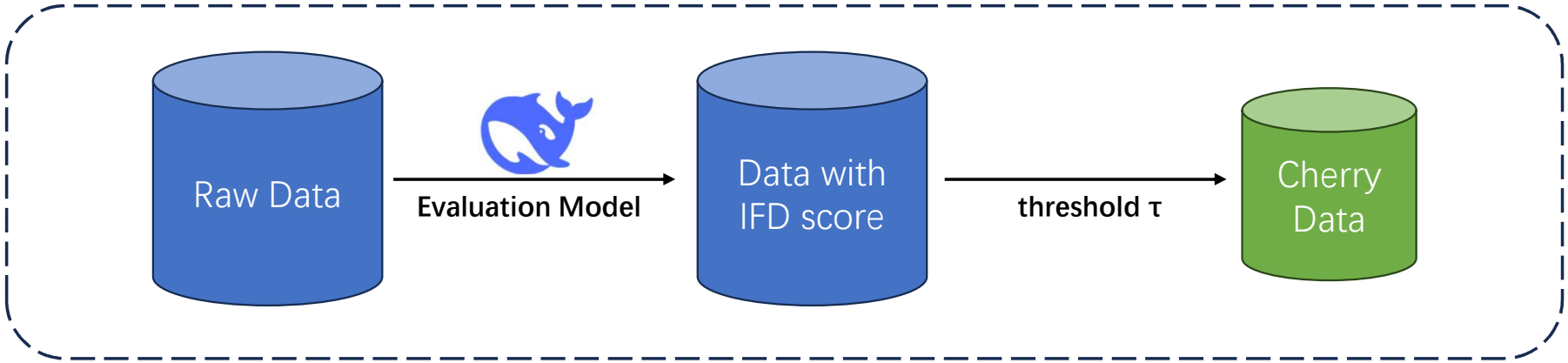
1. Firstly, we employ a filtering mechanism based on the **Instruction-Following Difficulty (IFD) score to selectively retain high-quality data samples** from the training dataset. By discarding low-quality samples, we effectively reduce the training data volume, thereby accelerating the training process.

2. Secondly, we introduce a novel **dynamic packing tokenization strategy**, which not only enhances the training speed but also improves the model's performance. Concurrently, our dynamic packing approach reduces the memory footprint, enabling more efficient utilization of available GPU memory.

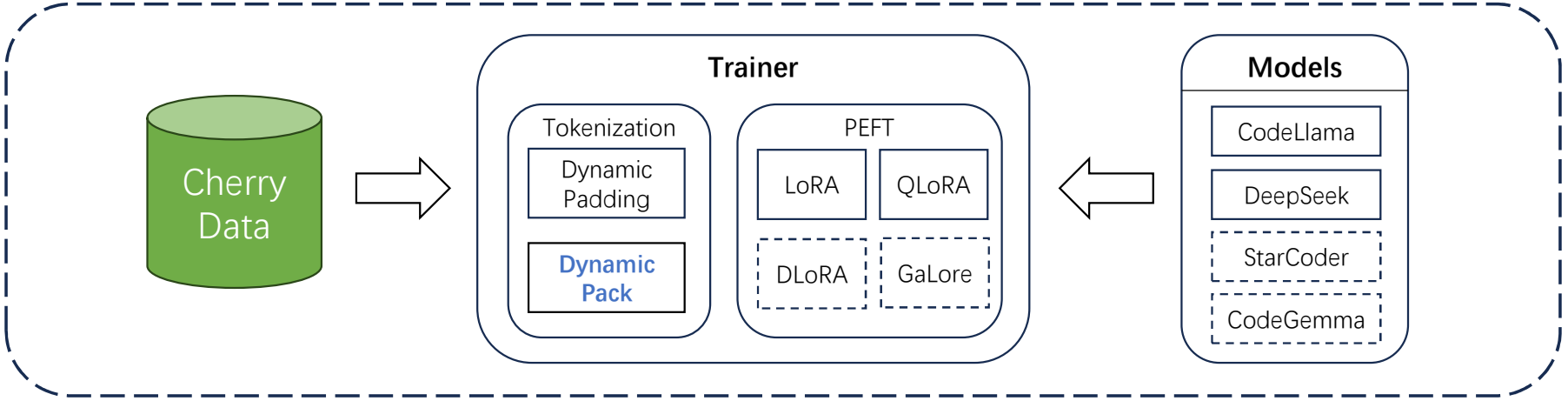
3. Thirdly, our framework supports efficient training of various mainstream code models, substantially lowering the computational requirements. Notably, **our framework facilitates training these models on consumer-grade graphics cards**, making large-scale code model training accessible to a broader range of users with limited computational resources.

CodeAcT

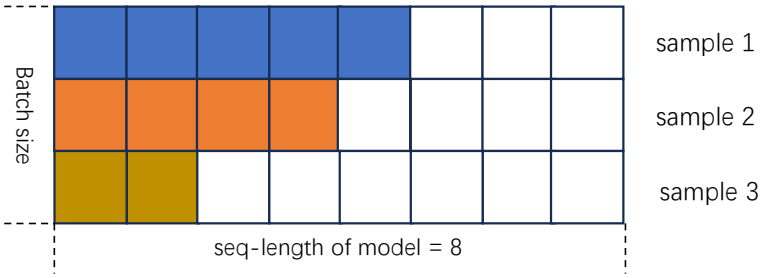
1. Data Selection



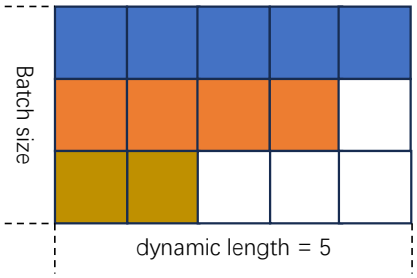
2. Parameter-Efficient Fine-Tuning



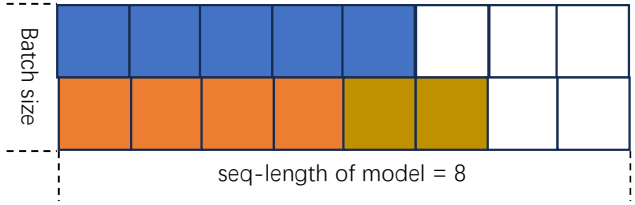
Dynamic Pack



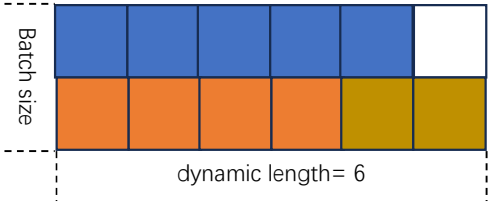
(a) Padding



(b) Dynamic Padding



(c) Pack (MFTCoder)



(d) Dynamic Pack (Ours)

Experiment

model	size	dataset	Train time	HumanEval pass@1	Eval time	备注
baseline						
CL-base	7B	-	-	31.7%	-	解码策略使用贪婪搜索
	13B	-	-	36.0%	-	
训练设置：4卡A100, [LoRA，r=32，alpha=64], [bs=8, as=8], 解码策略使用贪婪搜索						
CL-base	7B	OSS-75k	4h53min	?	?	dynamic pad，原始数据训练，数据量75197
			4h8min	36.6%	12min	dynamic pad，筛选出IFD得分在[0.5, 1.0]的数据，数据占比约84%，数据量62,922
			3h3min	38.4%	9min	dynamic pack，筛选出IFD得分在[0.5, 1.0]的数据，数据占比约84%，数据量62,922
			1h30min	**38.4%**	10min	dynamic pack，筛选出IFD得分在[0.7, 1.0]的数据，数据占比约**39%**，数据量**29,026**
			35min	34.1%	16min	dynamic pack，筛选出IFD得分在[0.8, 1.0]的数据，数据占比约14%，数据量10,380
训练设置：4卡A100, [LoRA，r=32，alpha=64], [bs=16, as=4], 解码策略使用贪婪搜索						
CL-base	7B	Evol-80k	7h7min	36.0%	16min	dynamic pad，原始数据训练，数据量78,258
			5h57min	?	?	dynamic pad，筛选出IFD得分在[0.5, 1.0]的数据，数据占比约80%，数据量62,432
			2h51min	37.2%	13min	dynamic pack，筛选出IFD得分在[0.5, 1.0]的数据，数据占比约80%，数据量62,432
			2h23min	**37.8%**	14min	dynamic pack，筛选出IFD得分在[0.7, 1.0]的数据，数据占比约60%，数据量47,208
			1h44min	**38.4%**	17min	dynamic pack，筛选出IFD得分在[0.8, 1.0]的数据，数据占比约**39%**，数据量**30,544**

Awards

2024年度第二届“启智杯”机器视觉设计大赛

优秀奖

在2024年度第二届“启智杯”机器视觉设计大赛中，**模型大师**团队积极参与并表现优异，荣获**算法赛道优秀奖**。

特发此证，以兹鼓励。

参赛团队 3 名成员如下。

姓名	高校
吕伟杰	南京航空航天大学
黄汇川	南京航空航天大学
顾建成	南京航空航天大学

浙江省仪器仪表学会　浙江省自动化学会　杭州海康机器人股份有限公司

2024年4月

Thanks