

Table R1: Pass@1 results on HumanEval (HumanEval+) and MBPP (MBPP+) benchmarks. *The column of FT Tokens (Source) specifies the tokens and origin of fine-tuning (FT) data used to train each model.*

Model	Params	Base Model	FT Tokens (Source)	HumanEval (+)	MBPP (+)	Average (+)
<i>Closed-source Models</i>						
GPT-3.5-Turbo	-	-	-	72.6 (65.9)	81.7 (69.4)	77.2 (67.7)
GPT-4-Turbo	-	-	-	<b>85.4 (81.7)</b>	<b>83.0 (70.7)</b>	<b>84.2 (76.2)</b>
<i>Open-source Models</i>						
Llama 2-Chat	70B	Llama 2	-	31.7 (26.2)	52.1 (38.6)	41.9 (32.4)
CodeLlama-Python	70B	Llama 2	-	57.9 (50.0)	72.4 (52.4)	65.2 (51.2)
CodeLlama-Instruct	70B	CodeLlama	5B (Llama Generation)	<b>65.2 (58.5)</b>	<b>73.5 (55.1)</b>	<b>69.4 (56.8)</b>
CodeLlama-Python	34B	Llama 2	-	51.8 (43.9)	67.2 (50.4)	59.5 (47.2)
WizardCoder-CL	34B	CodeLlama-Python	80M (GPT Generation)	73.2 (56.7)	73.2 (51.9)	73.2 (54.3)
DeepSeek-Coder-Instruct	33B	DeepSeek-Coder-Base	2B (GitHub Crawling)	<b>78.7 (67.7)</b>	<b>78.7 (59.7)</b>	<b>78.7 (63.7)</b>
StarCoder	15B	-	-	34.1 (33.5)	55.1 (43.4)	44.6 (38.5)
CodeLlama-Python	13B	Llama 2	-	42.7 (36.6)	61.2 ( <b>45.6</b> )	52.0 (41.1)
WizardCoder-SC	15B	StarCoder	80M (GPT Generation)	<b>51.9 (45.7)</b>	<b>61.9 (44.9)</b>	<b>56.9 (45.3)</b>
Llama 2	7B	-	-	14.0 (10.4)	26.1 (17.5)	20.1 (14.0)
StarCoder	7B	-	-	24.4 (21.3)	33.1 (29.2)	28.8 (25.3)
CodeLlama-Python	7B	Llama 2	-	37.8 (33.5)	57.6 (42.4)	47.7 (38.0)
WizardCoder-CL	7B	CodeLlama-Python	80M (GPT Generation)	48.2 (42.1)	56.6 (42.4)	52.4 (42.3)
DeepSeek-Coder-Base	6.7B	-	-	47.6 (41.5)	70.2 (53.6)	58.9 (47.6)
Magicode-CL	7B	CodeLlama-Python	90M (GPT Generation)	60.4 (49.4)	64.2 (46.1)	62.3 (47.8)
MagicodeS-CL	7B	CodeLlama-Python	240M (GPT Generation)	70.7 (60.4)	68.4 (49.1)	69.6 (54.8)
Magicode-DS	6.7B	DeepSeek-Coder-Base	90M (GPT Generation)	66.5 (55.5)	75.4 (55.6)	71.0 (55.6)
DeepSeek-Coder-Instruct	6.7B	DeepSeek-Coder-Base	2B (GitHub Crawling)	73.8 (69.5)	72.7 (55.6)	73.3 (62.6)
MagicodeS-DS	6.7B	DeepSeek-Coder-Base	240M (GPT Generation)	76.8 (65.2)	75.7 (56.1)	76.3 (60.7)
<i>AlchemistCoder-L (ours)</i>	7B	Llama 2	200M (GPT Harmonization)	56.7 (52.4)	54.5 (49.6)	55.6 (51.0)
<i>AlchemistCoder-CL (ours)</i>	7B	CodeLlama-Python	200M (GPT Harmonization)	74.4 (68.3)	68.5 (55.1)	71.5 (61.7)
<i>AlchemistCoder-DS (ours)</i>	6.7B	DeepSeek-Coder-Base	200M (GPT Harmonization)	<b>79.9 (75.6)</b>	<b>77.0 (60.2)</b>	<b>78.5 (67.9)</b>

**Instruction:**  
Change the color of the text to blue.  
## Input:  
Text: The quick brown fox

**Response:**  
To change the color of the text to blue, you can use CSS. Here's an example of how you can achieve this:

```
<style>  
  .blue-text {  
    color: blue;  
  }  
</style>  
<p class="blue-text">The quick brown fox</p>
```

By applying the "blue-text" class to the paragraph element, the text will be displayed in blue color.

**Inherent Conflict:**  
*Missing Language Declaration.* The instruction does not specify the programming language, while the response assumes HTML/CSS and provides an implementation accordingly.

**AlchemistPrompt-customized Instruction:**  
Assume we are working with a HTML document and have the permissions to modify the CSS styles associated with it.  
Change the color of the text to blue.  
## Input:  
Text: The quick brown fox

**Instruction:**  
Find the minimum of two given numbers.  
## Input: 4, 6  
## Output: The minimum number is 4

**Response:**  
\*Notebook Format:\*  
\*\*Reasoning:\*\*  
To find the minimum of two given numbers, we can use an if-else statement. We will compare the two numbers, assign the smaller value to a variable called "minimum\_number", and print out it.  
\*\*Code:\*\*

```
num1, num2 = 4, 6  
if num1 < num2:  
    minimum_number = num1  
else:  
    minimum_number = num2  
print("The minimum number is", minimum_number)
```

**Execution Results:\*\***  
The minimum number is 4

**Inherent Conflict:**  
*Format Exception.* The response introduces a notebook format not specified by the instruction.

**AlchemistPrompt-customized Instruction:**  
Complete the simple task in notebook format, including reasoning, code, and execution results.  
Find the minimum of two given numbers.  
## Input: 4, 6  
## Output: The minimum number is 4

**Instruction (Source #1&2):**  
Replace all words with 5 letters or more that start with a vowel, and end with a consonant, with "\*\*\*\*\*" in the given text.  
"The quick brown fox jumps over the lazy dog"

**Response (Source #1):**  
Sure! Here is the text with the specified replacements:  
"The quick brown fox jumps \*\*\*\*\* the lazy dog"

**Response (Source #2):**  
Here's a Python function that achieves this:

```
import re  
def replace_words(text):  
    pattern = r'\b[aeiouAEIOU]\w{3,}[^aeiouAEIOU]\b'  
    result = re.sub(pattern, '*****', text)  
    return result
```

text = "The quick brown fox jumps over the lazy dog"  
print(replace\_words(text))

**Inherent Conflict:**  
*Response Format Discrepancy.* For similar instructions, Source #1 provides the result directly, while Source #2 provides the implementation code, which may lead to confusion in understanding the instructions.

**AlchemistPrompt-customized Instruction (Source #2):**  
In this task, you are required to create a Python program that scans through a given text and replaces specified words.  
Replace all words with 5 letters or more that start with a vowel, and end with a consonant, with "\*\*\*\*\*" in the given text.  
"The quick brown fox jumps over the lazy dog"

Figure R1: Examples of inherent conflicts (e.g., various styles and quality) within multi-source code corpora. By applying *AlchemistPrompt*-customized instructions that are more consistent with the responses, the diversity from multiple sources can be effectively managed and utilized, thereby improving the quality of our fine-tuning data and the instruction-following capabilities of the fine-tuned models.