The Curious Case of Code Prompts

Li Zhang*, Liam Dugan*, Hainiu Xu*, Chris Callison-Burch

Penn Engineering

Paper:

TLDR

- **Previously**, papers such as (Madaan et al., 2022), (Wang et al., 2022), (Dong et al., 2022), (Bi et al., 2023), and (Zhang et al., 2023) found that prompting LLMs with **code**-like **prompts** leads to better performance.

Text Prompt

```
You are trying to {goal}. You
need to do two things:
(a) {step0}
(b) {step1}
The first thing to do
is {first}
```

Code Prompt (vanilla)

```
input0 = "Given a goal and two steps,
predict the correct order to do the
steps to achieve the goal"
input1 = "{goal}"
step0 = "{step0}"
step1 = "{step1}"
label = [{first},{second}]
```

Code Prompt (VI - var identifier)

```
instructions = "Given a goal and two
steps, predict the correct order to do
the steps to achieve the goal"
goal = "{goal}"
step0 = "{step0}"
step1 = "{step1}"
order_of_exec = [{first},{second}]
```

Code Prompt (VIC - var identifier + comments)

```
"""Given a goal and two steps, predict the correct
order to do the steps to achieve the goal"""

# The goal that someone is trying to achieve
goal = "{goal}"

# One of the steps that needs to be taken
step0 = "{step0}"

# Another one of the steps that need be taken
step1 = "{step1}"

# The list of correct order of those two steps
order_of_exec = [{first},{second}]
```

Code Prompt (CVIC - class + var identifier + comments)

```
import order_steps
class Event:
  """Given a goal and two steps, predict the correct
  order to do the steps to achieve the goal"""
  def __init__(self, goal, step0, step1):
    self.goal = goal # The goal someone is trying to accomplish
    self.step0 = step0 # One of the steps that need be taken
    self.step1 = step1 # Another step that need be taken
  def get_order_of_steps(self):
    # Output a list of correct order of the two steps to be taken
    return order_steps(self.goal, self.step0, self.step1)

event = Event(goal="{goal}", step0="{step0}", step1="{step1}")
assert(event.get_order_of_steps == [{first},{second}])
```

- Those works focus on a small subset of reasoning tasks. Intuitively, code prompts may better ellicit LLMs' reasoning ability.
- In this work, we ask: **are code prompts universally good?** No.

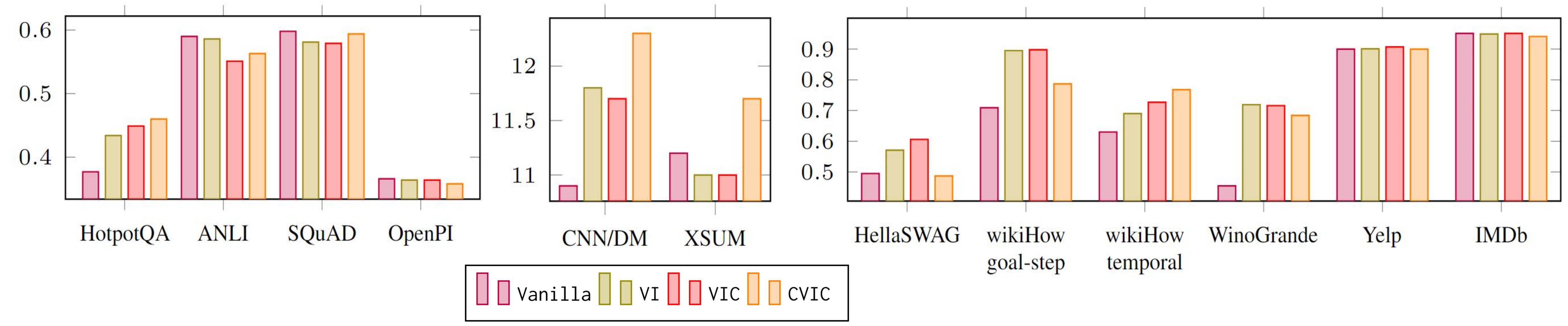
Datasets

- We select a dozen datasets from popular NLP tasks.

Dataset	Task Category	Num. Eval Examples	Metric	Origin
HellaSwag	Commonsense Reasoning	1000 / 10042	Accuracy	Zellers et al. (2019)
wikiHow Goal-Step	Commonsense Reasoning	1000 / 1073	Accuracy	Zhang et al. (2020)
wikiHow Temporal	Commonsense Reasoning	1000 / 3100	Accuracy	Zhang et al. (2020)
WinoGrande	Commonsense Reasoning	1000 / 1767	Accuracy	Sakaguchi et al. (2021)
OpenPI	Commonsense Reasoning	111 / 111	ROUGE-F1	Tandon et al. (2020)
ANLI	Natural Language Inference	1000 / 3000	Accuracy	Nie et al. (2020)
Yelp	Sentiment Analysis	1000 / 10000	Pearson's r	Zhang et al. (2015)
IMDb	Sentiment Analysis	1000 / 25000	Accuracy	Maas et al. (2011)
HotpotQA	Question Answering	1000 / 7405	Macro-F1	Yang et al. (2018)
SQuAD	Question Answering	1000 / 11873	Macro-F1	Rajpurkar et al. (2018)
CNN/Daily Mail	Summarization	1000 / 13368	ROUGE-2	Nallapati et al. (2016)
XSUM	Summarization	1000 / 11332	ROUGE-2	Narayan et al. (2018)

Results & Analysis

- We consider **`code-davinci-002`** and **`text-davinci-002`**, two GPT3.5 models; and **`davinci`**, the base model without any code pretraining.
- **Question #1: what kind of code prompt is better?**



- Answer #1: no clear trend, but variable identifier and comments help.
- **Question #2: is code prompt better than text prompt?**

Dataset	Metric	davinci +Text	davinci +Code	davinci Δ	code-002 +Text	code-002 +Code	code-002 Δ	text-002 +Text	text-002 +Code	text-002 Δ
Hellaswag	Accuracy	0.321	0.307	-0.014	0.652	0.606	-0.046	0.717	0.773	+0.046
wikiHow goal-step	Accuracy	0.347	0.302	-0.045	0.924	0.898	-0.026	0.919	0.915	-0.004
wikiHow temporal	Accuracy	0.495	0.532	+0.037	0.622	0.727	+0.105	0.688	0.761	+0.073
Yelp	Pearson ρ	0.913	0.896	-0.017	0.924	0.907	-0.017	0.919	0.904	-0.015
IMDb	Accuracy	0.872	0.935	+0.063	0.945	0.951	+0.006	0.940	0.952	+0.012
WinoGrande	Accuracy	0.513	0.500	-0.013	0.607	0.716	+0.109	0.628	0.726	+0.098
ANLI	Accuracy	0.333	0.360	+0.027	0.562	0.551	-0.011	0.504	0.557	+0.053
HotpotQA	Macro-F1	-	-	-	0.470	0.449	-0.021	0.490	0.350	-0.140
SQuAD	Macro-F1	0.482	0.466	-0.016	0.604	0.579	-0.025	0.670	0.656	-0.014
OpenPI	ROUGE-F1	-	-	-	37.33	36.36	-0.970	35.60	31.30	-4.300
CNN/Daily Mail	ROUGE-2	9.28	9.13	-0.150	11.74	11.67	-0.070	13.63	13.55	-0.080
XSUM	ROUGE-2	9.38	6.83	-2.550	14.51	11.03	-3.580	14.48	13.26	-1.220

- Answer #2: Overall, code prompts do not consistently outperform text prompts, nor do they underperform, even for reasoning tasks.
- Also note: LLM with supervised textual finetuning, **`text-davinci-002`**, is just as capable of working with code prompts, compared to **`code-davinci-002`**

Future work must continue to decide whether to use **code prompts** or **text prompts** on a case-by-case basis.