Taming of a stochastic parrot

STAAR ERC Summer Workshop 2024

Shin Yoo | COINSE@KAIST | 2024.07.08

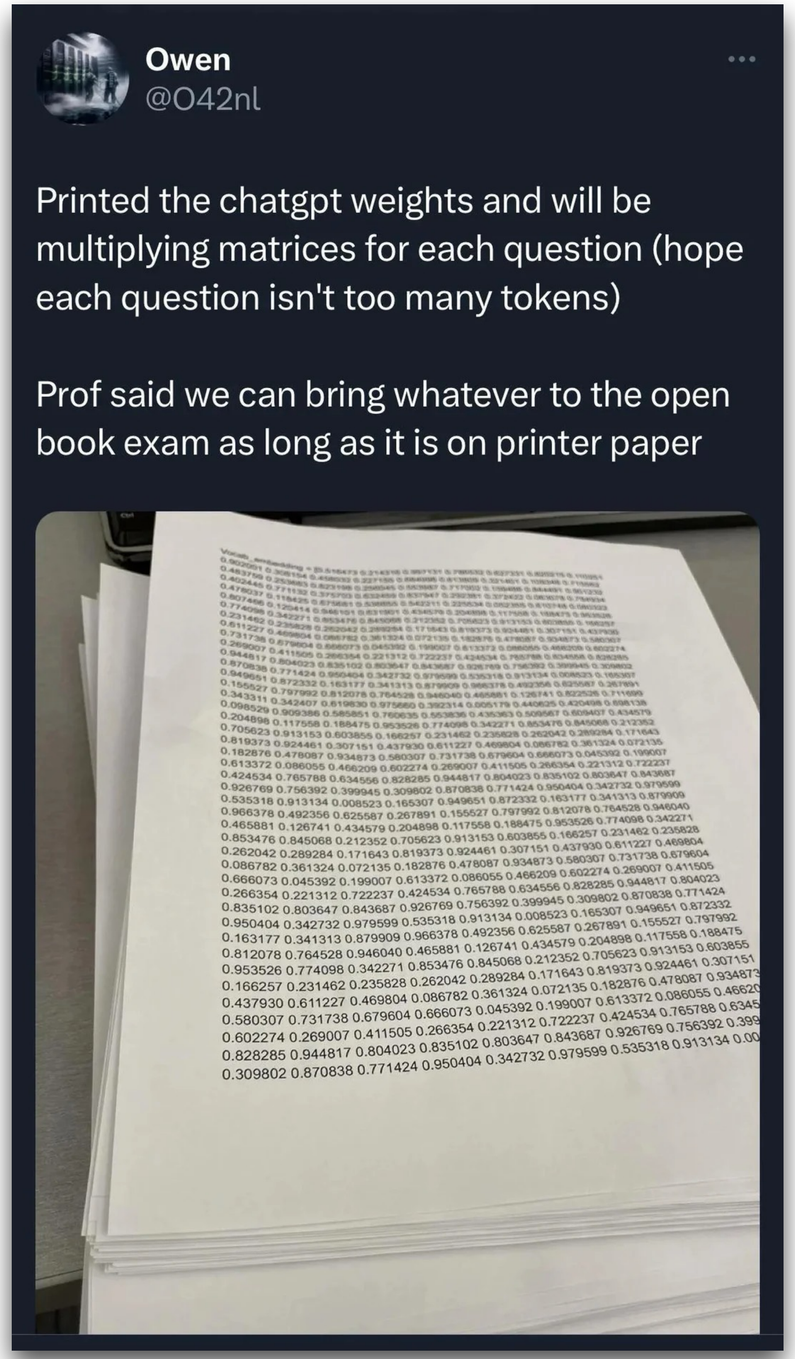
漢
The Chinese Room

A Thought Experiment

John Searle, “Mind, Brains, and Programs” in 1980

- Suppose we have a computer program that behaves as if it understands Chinese language.
- You are in a closed room with the AI program source code.
- Someone passes a paper with Chinese characters written on it, into the room.
- You use the source code as instruction to generate the response to the input, and sends the response out of the room.
- Do you understand Chinese language, or not?

“And we’re talking about this because…”



(Obviously we are all a bit like this now)

Stochastic Parrot?

- Among other risks, authors ask whether LLMs actually “understand” anything.
- What do you think?
- The internal design is clearly a statistical language model, i.e., it says what is the most likely, not what is the correct.

Check for updates

On the Dangers of Stochastic Parrots: Can Language Models Be Too Big? 🦜

Emily M. Bender*
ebender@uw.edu
University of Washington
Seattle, WA, USA

Timnit Gebru*
timnit@blackinai.org
Black in AI
Palo Alto, CA, USA

Angelina McMillan-Major
aymm@uw.edu
University of Washington
Seattle, WA, USA

Shmargaret Shmitchell
shmargaret.shmitchell@gmail.com
The Aether

ABSTRACT

The past 3 years of work in NLP have been characterized by the development and deployment of ever larger language models, especially for English. BERT, its variants, GPT-2/3, and others, most recently Switch-C, have pushed the boundaries of the possible both through architectural innovations and through sheer size. Using these pretrained models and the methodology of fine-tuning them for specific tasks, researchers have extended the state of the art on a wide array of tasks as measured by leaderboards on specific benchmarks for English. In this paper, we take a step back and ask: How big is too big? What are the possible risks associated with this technology and what paths are available for mitigating those risks? We provide recommendations including weighing the environmental and financial costs first, investing resources into curating and carefully documenting datasets rather than ingesting everything on the web, carrying out pre-development exercises evaluating how the planned approach fits into research and development goals and supports stakeholder values, and encouraging research directions beyond ever larger language models.

CCS CONCEPTS

• **Computing methodologies** → **Natural language processing**.

ACM Reference Format:
Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the Dangers of Stochastic Parrots: Can Language Models Be Too Big? 🦜. In *Conference on Fairness, Accountability, and Transparency (FAccT '21), March 3–10, 2021, Virtual Event, Canada.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3442188.3445922

1 INTRODUCTION

One of the biggest trends in natural language processing (NLP) has been the increasing size of language models (LMs) as measured by the number of parameters and size of training data. Since 2018 alone, we have seen the emergence of BERT and its variants [39, 70, 74, 113, 146], GPT-2 [106], T-NLG [112], GPT-3 [25], and most recently Switch-C [43], with institutions seemingly competing to produce ever larger LMs. While investigating properties of LMs and how they change with size holds scientific interest, and large LMs have shown improvements on various tasks (§2), we ask whether enough thought has been put into the potential risks associated with developing them and strategies to mitigate these risks.

We first consider environmental risks. Echoing a line of recent work outlining the environmental and financial costs of deep learning systems [129], we encourage the research community to prioritize these impacts. One way this can be done is by reporting costs and evaluating works based on the amount of resources they consume [57]. As we outline in §3, increasing the environmental and financial costs of these models doubly punishes marginalized communities that are least likely to benefit from the progress achieved by large LMs and most likely to be harmed by negative environmental consequences of its resource consumption. At the scale we are discussing (outlined in §2), the first consideration should be the environmental cost.

Just as environmental impact scales with model size, so does the difficulty of understanding what is in the training data. In §4, we discuss how large datasets based on texts from the Internet overrepresent hegemonic viewpoints and encode biases potentially damaging to marginalized populations. In collecting ever larger datasets we risk incurring documentation debt. We recommend mitigating these risks by budgeting for curation and documentation at the start of a project and only creating datasets as large as can be sufficiently documented.

As argued by Bender and Koller [14], it is important to understand the limitations of LMs and put their success in context. This not only helps reduce hype which can mislead the public and researchers themselves regarding the capabilities of these LMs, but might encourage new research directions that do not necessarily depend on having larger LMs. As we discuss in §5, LMs are not performing natural language understanding (NLU), and only have success in tasks that can be approached by manipulating linguistic form [14]. Focusing on state-of-the-art results on leaderboards without encouraging deeper understanding of the mechanism by which they are achieved can cause misleading results as shown

*Joint first authors

FAccT '21, March 3–10, 2021, Virtual Event, Canada
ACM ISBN 978-1-4503-8309-7/21/03.

John: Hi, nice to meet you. How are you?

Mary: I’m ____, _____ ___. ___ ___?

a) fine, thank you. And you?

b) okay, I guess. But why?

Python: for _ __ _____ …

a) i in range

b) (int i = 0;

Java: for _ ___ _ _ _ _ …

a) i in range

b) (int i = 0;

```
def SieveOfEratosthenes(num):

  a) prime = [True for i in range(num+1)]…

  b) arr = re.findall(r'[0-9]+', word)…
```

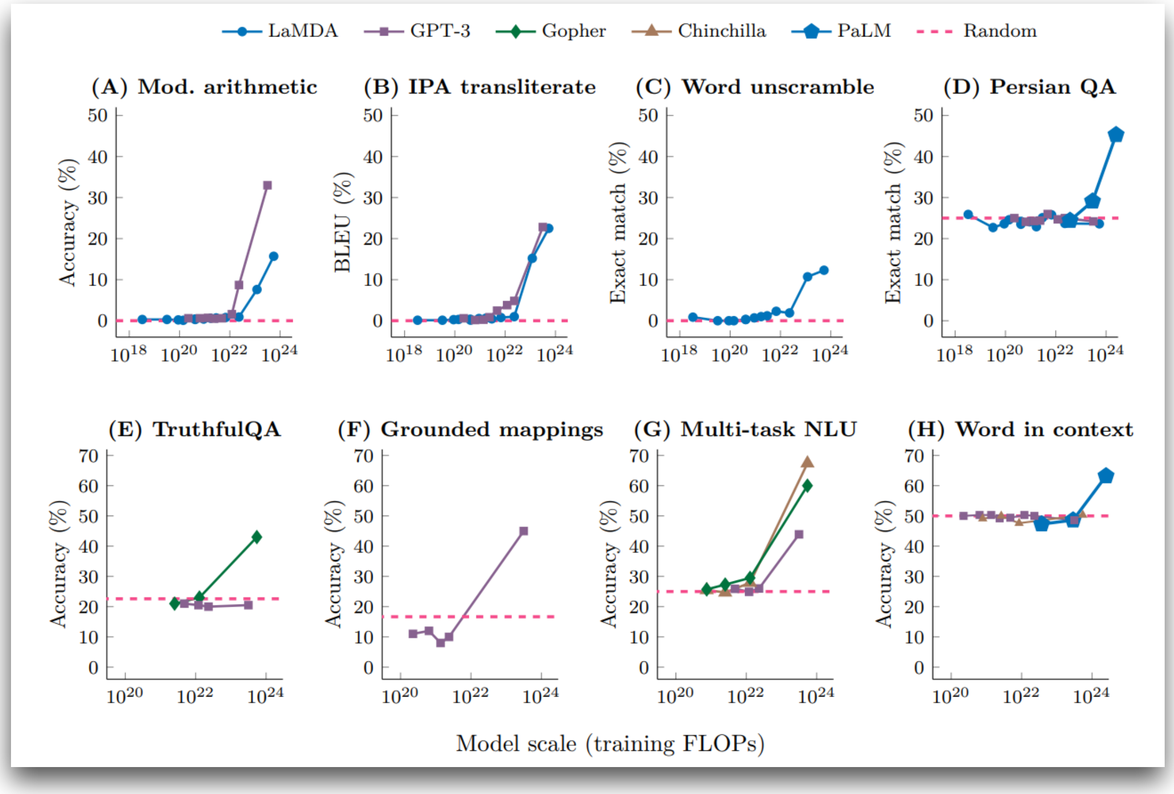
Large Language Model

(really, a very large statistical language model)

- Mainly Transformer-based DNNs that are trained to be an auto-regressive language model, i.e., given a sequence of tokens, it repeatedly tries to predict the next token.
- The biggest hype in SE research right now with an **explosive** growth, because:
  - **Emergent behaviour** leading to very attractive properties such as in-context learning, Chain-of-Thoughts, or PAL
  - They **seem to** get the **semantics** of the code and **work across natural and programming language**

What is an Emergent Behavior?

- Above certain size, LLMs change their behavior in interesting ways
- The point of change in slope is referred to as “breaks”



Caballero et al., https://arxiv.org/abs/2210.14891

Chain-of-Thoughts

Wei et al., https://arxiv.org/abs/2201.11903

- Underneath, LLMs are doing autocompletion, not any other type of reasoning: they appear to be capable of rational inference because the corpus they are trained with includes traces of logical reasoning.
- So, **conditioning** the model (with the context) to be more precise about the reasoning steps can result in generation of more accurate reasoning steps.
  - Add “Let’s think in step by step” at the end of every prompt (https://arxiv.org/abs/2205.11916) 🙃 😐 🫠

Chain-o

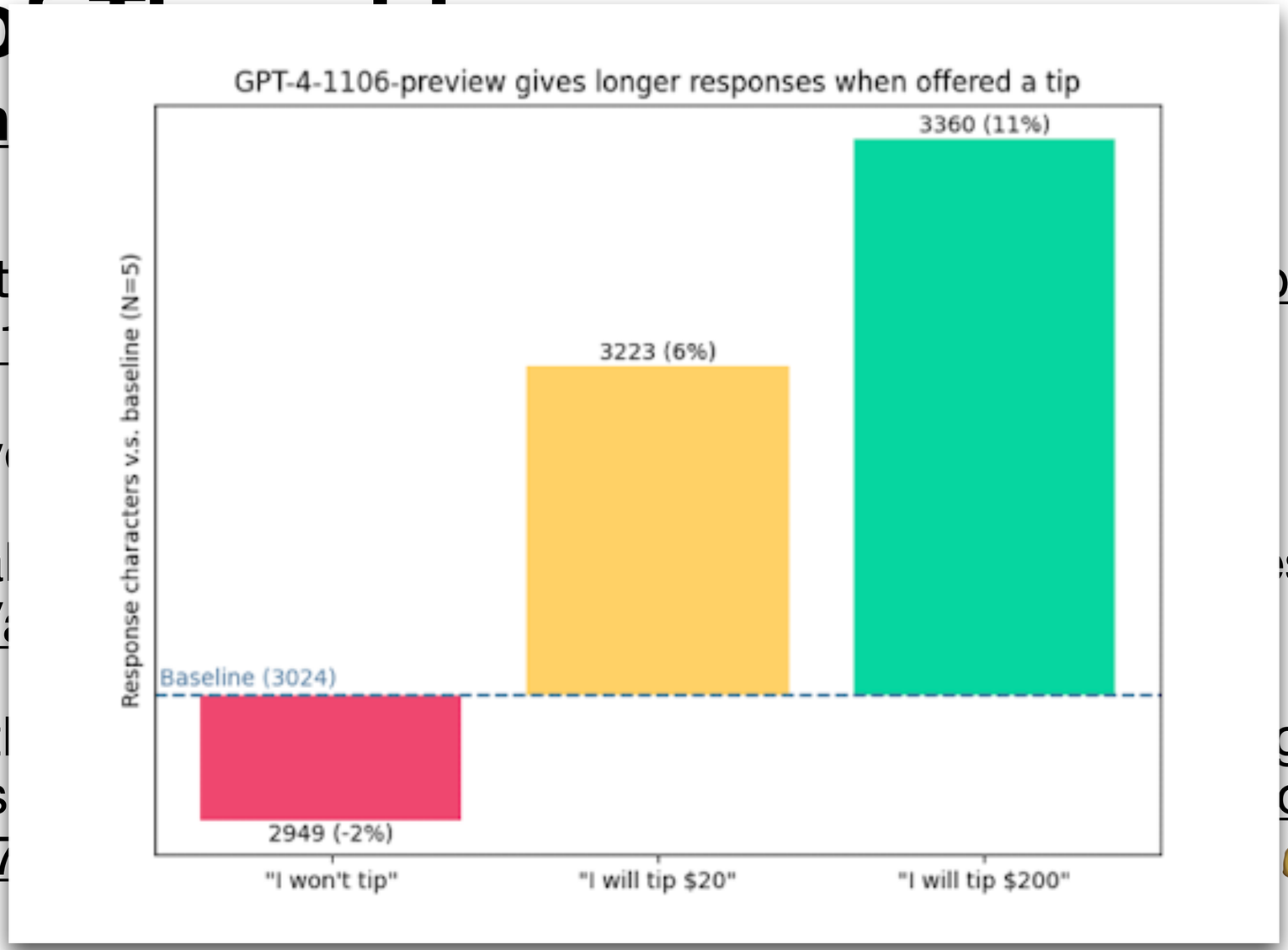
Wei et al., h

- Add “Let’s t
- We have ev
  - If you ma
  - Apparentl

ps://arxiv.org/abs/2205.11

es (https://arxiv.org/a

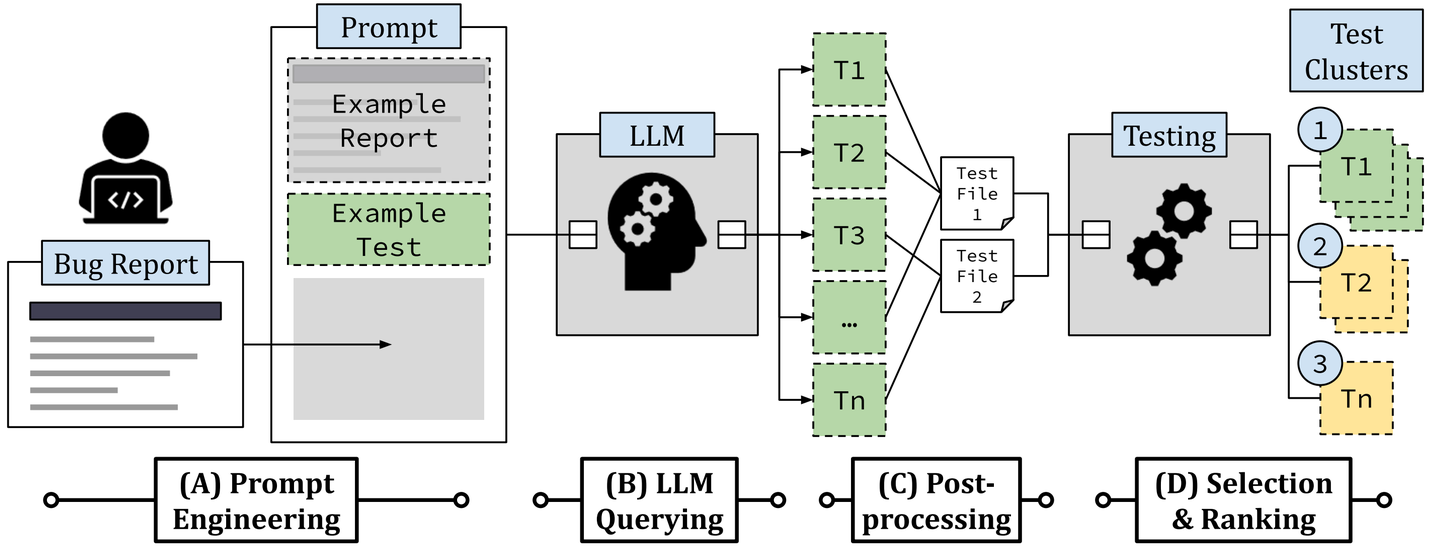
ge tip produces oogel/status/1730726



“Okay, it talks like a human and can answer some questions. But why SE?”

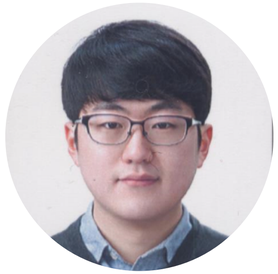
LLMs seemingly handle semantics across NL/PL barrier

LLM-based Bug Reproduction (Kang, Yoon & Yoo, ICSE 2023)



Title	assertContainsIgnoringCase fails to compare i and I in tr_TR locale
	See org.assertj.core.internal.Strings#assertContainsIgnoringCase [url] I would suggest adding [url] verification to just ban toLowerCase(), toUpperCase() and other unsafe methods: #2664

```java
public void testIssue952(){
  Locale locale = new Locale("tr", "TR");
  Locale.setDefault(locale);
  assertThat("I").as("Checking in tr_TR locale")
  .containsIgnoringCase("i");
}
```

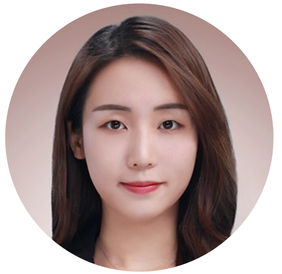


Sungmin Kang
(PhD Candidate)

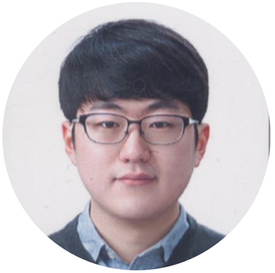
Juyeon Yoon
(PhD Candidate)

AutoFL: LLM based FL

Kang, An & Yoo (https://arxiv.org/abs/2308.05487)



Gabin An
(PhD Candidate)



Sungmin Kang
(PhD Candidate)

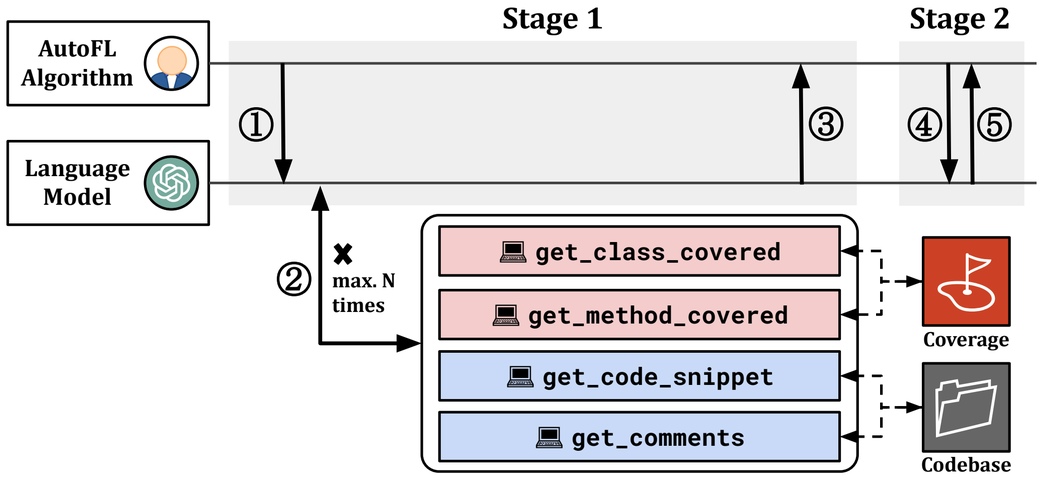


Figure 1: Diagram of AutoFL. Each arrow represents a prompt / response between components, with the circled numbers indicating the order of interactions. Function invocations are made at most N times, where N is a predetermined parameter of AutoFL.

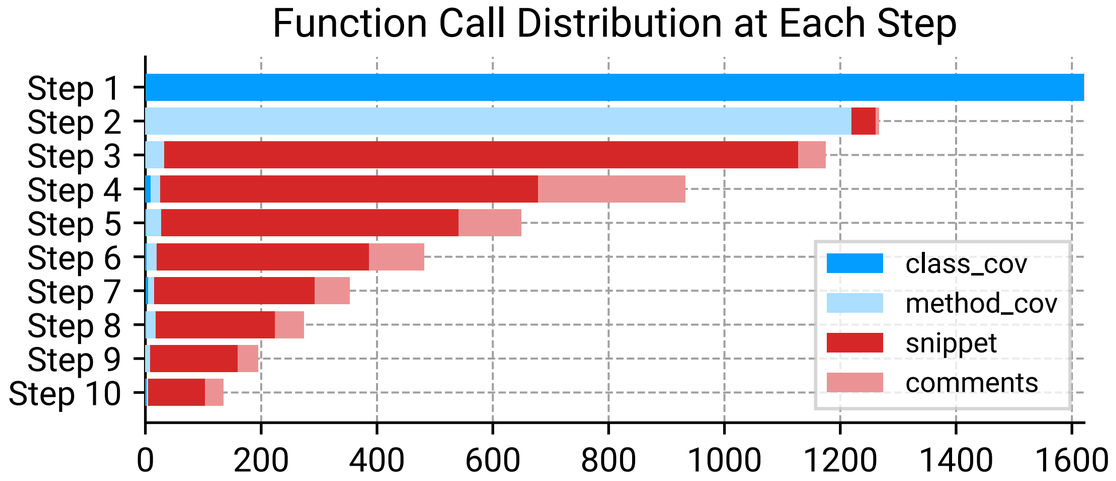
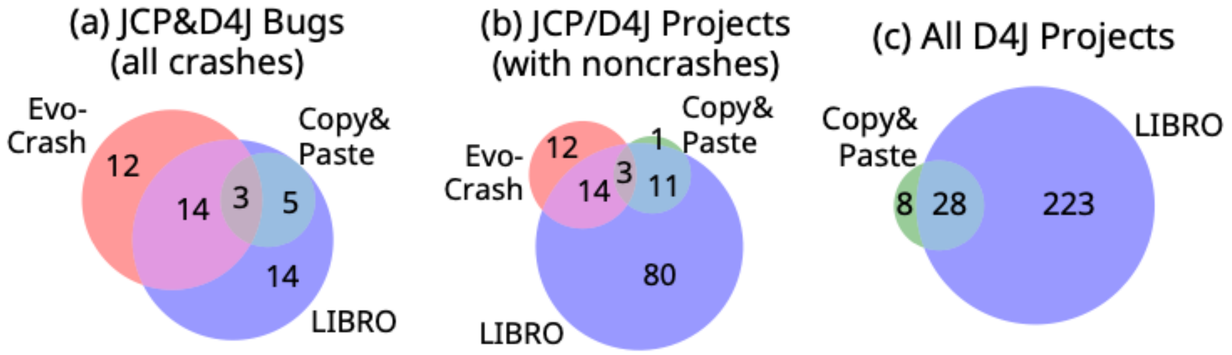


Figure 5: Function call frequency by step over all five runs of AutoFL. The total length at each step decreases as AutoFL can stop calling functions at any step; e.g. about 400 AutoFL processes stopped calling functions after the first step.



Libro Reproduction Results
(against of 750 Bugs)

Family	Technique	acc@1	acc@3	acc@5
Predicate Switching		42	99	121
Stack Trace		57	108	130
Slicing (frequency)		51	96	119
MBFL	MUSE	73	139	161
	Metallaxis	106	162	191
SBFL	Ochiai	122	192	**218**
	DStar	125	**195**	216
	SBFL-F	34	66	78
LLM-Based	LLM+Test	81	94	97
	AutoFL	**149**	180	194

AutoFL Evaluation Metric
(against of 353 Bugs)

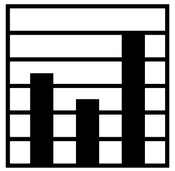
“Sounds like LLMs will solve all SE problems. Can we go home now?”

My (very personal) current attitude towards LLM…

- It is not probably reaching AGI anytime soon: in fact, I doubt a statistical language model can be a truly general intelligence (whatever that means!).
- BUT this parrot can do many interesting tricks (such as moving back and forth between code and natural language)! Why not make use of them, while maintaining a cautious, balanced skepticism?
- What can we do to tame the stochastic parrot?

Executability

Landscape Analysis



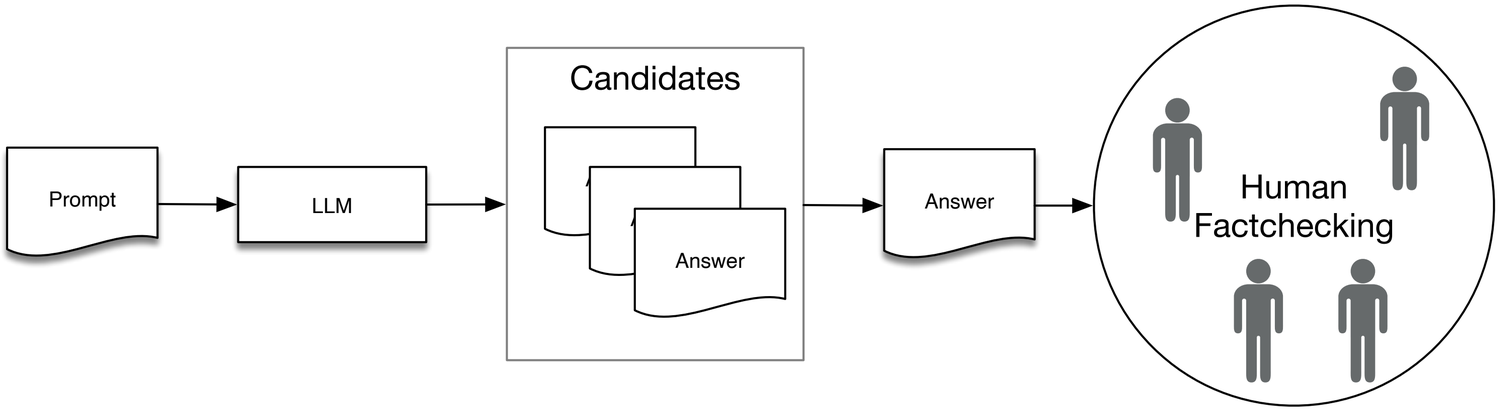
Mathematical Modelling

Executability

Code is a unique artifact because it executes.

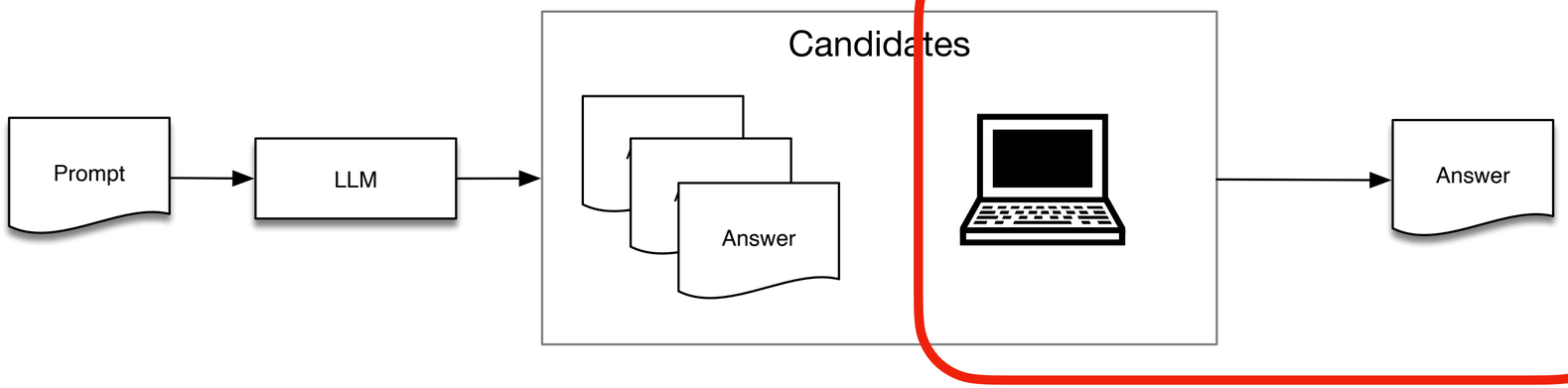
(And we've been doing dynamic analysis for a long time)

NL + LLM Pipeline



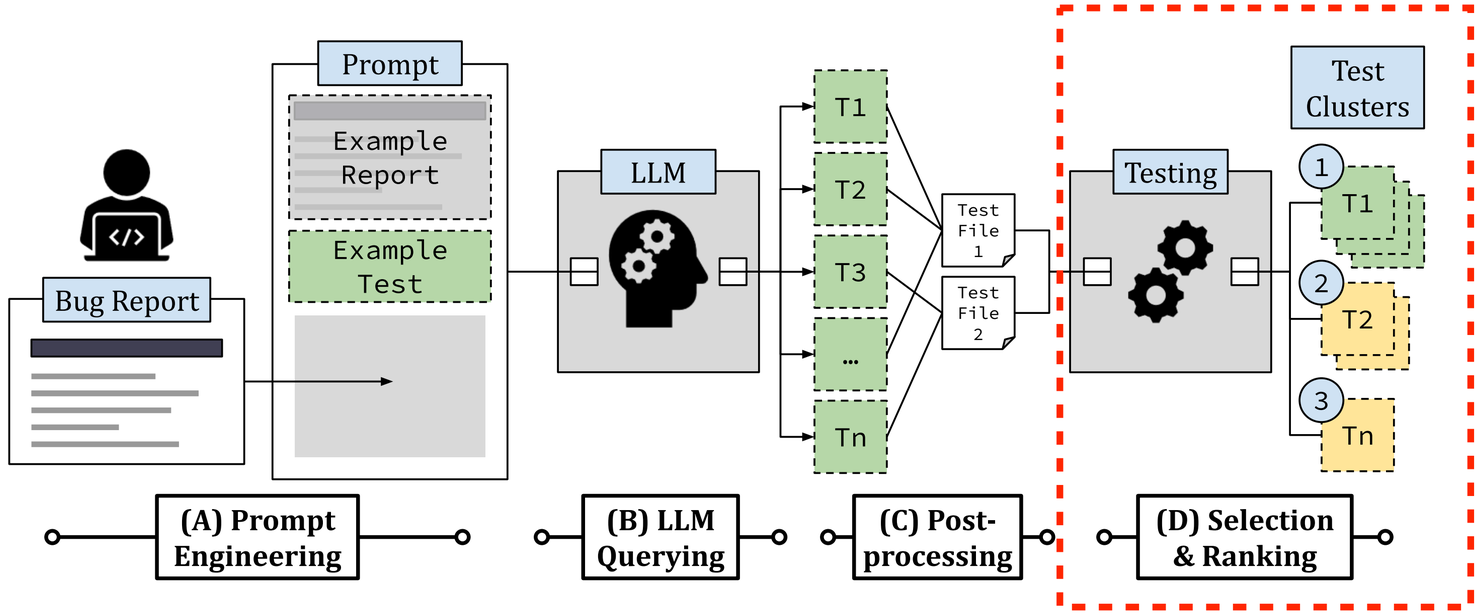
Isn't this testing? :)

PL/NL + LLM Pipeline

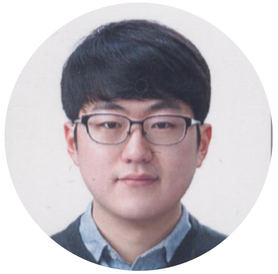


Execution enabling self-consistency

LLM-based Bug Reproduction (Kang, Yoon & Yoo, ICSE 2023)



- Any test that does not fail in the buggy version are filtered out
- Failure type and error messages are considered when clustering tests.



Dr. Sungmin Kang (KAIST)

Juyeon Yoon (PhD Candidate)

Agency through NL > PL > NL

Yoon et al., ICST 2024 (https://arxiv.org/abs/2311.08649)

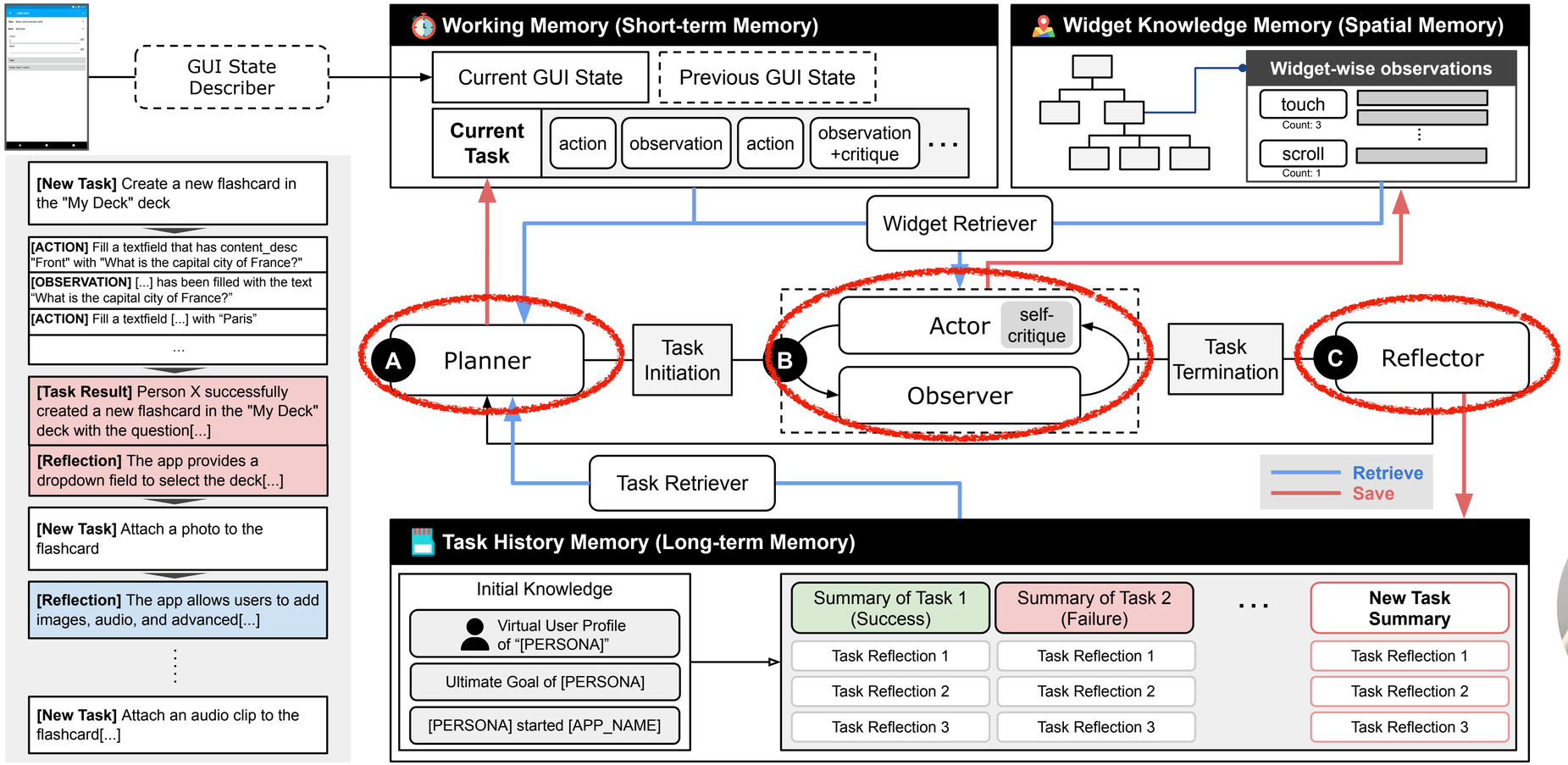


Fig. 1. Overview of DROIDAGENT with a task example.

Juyeon Yoon
(PhD Candidate)

Prof. Robert Feldt
(Chalmers)

Secondary Executability

- “What about LLM-generated artefacts that cannot be directly executed, for example, documentation?”
- Exploiting NL↔PL capability, we can create opportunities for secondary execution, i.e., generate something executable out of the non-excutable!
  - Caveat: this does add noise to the decision.

Bad Comments results in Bad Tests

Kang, Milliken & Yoo (https://arxiv.org/pdf/2406.14836)

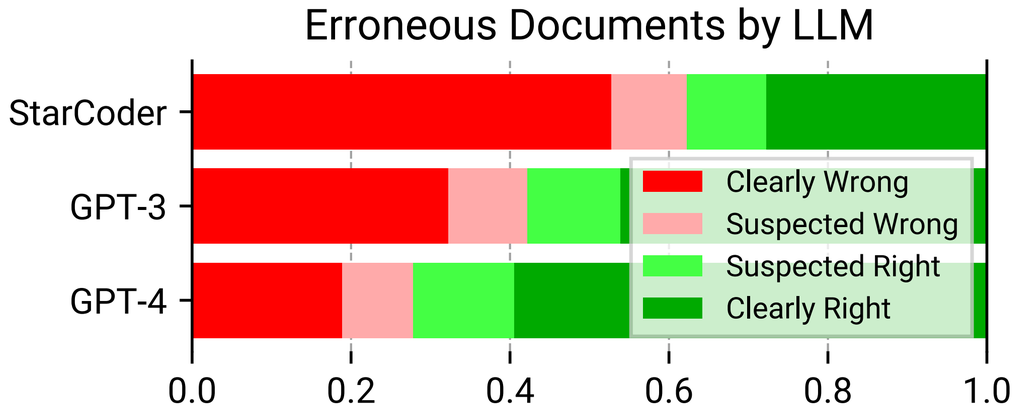


Figure 1: Comment factual accuracy by generating LLM.

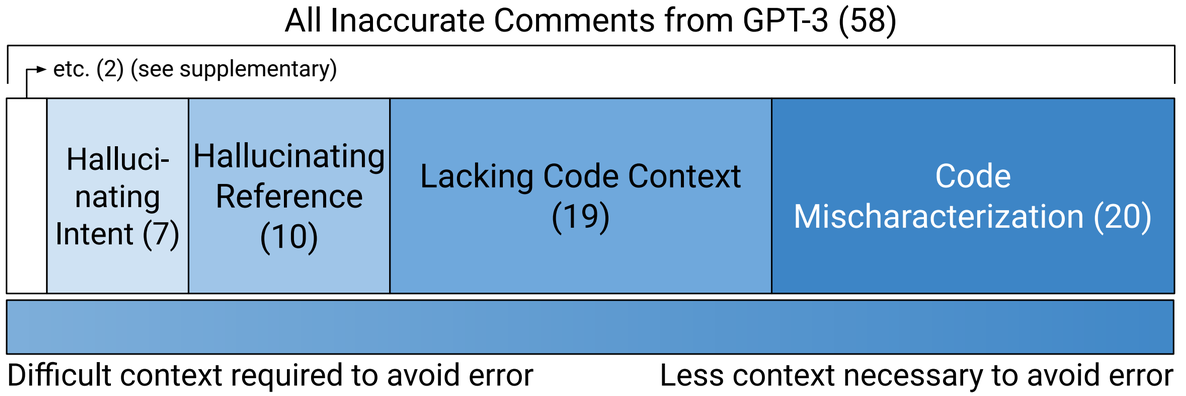
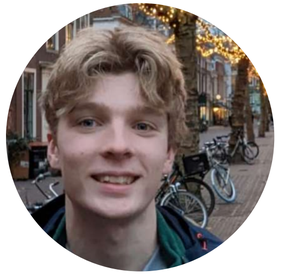


Figure 2: Diagram of error taxonomy for GPT-3 comments.

Dr. Sungmin Kang
(KAIST)

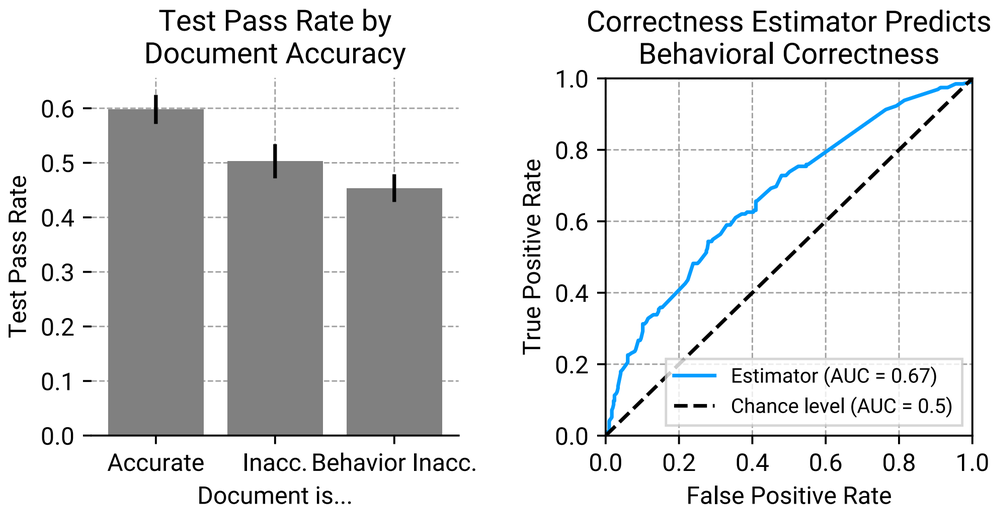


Louis Milliken
(MSc Candidate)

(we build up some Bayesian theory that says the probability of documentation being correct correlates with the number of passing tests generated based on the documents minus the number of failing tests generated based on the documents)

Test Pass Rates Correlates with Doc Quality

Kang, Milliken & Yoo (https://arxiv.org/pdf/2406.14836)



(a) Pass rate by accuracy, with 95% confidence intervals.

(b) ROC graph of correctness estimator with actual correctness.

Figure 4: Relationship between comment accuracy and suggested indicators.

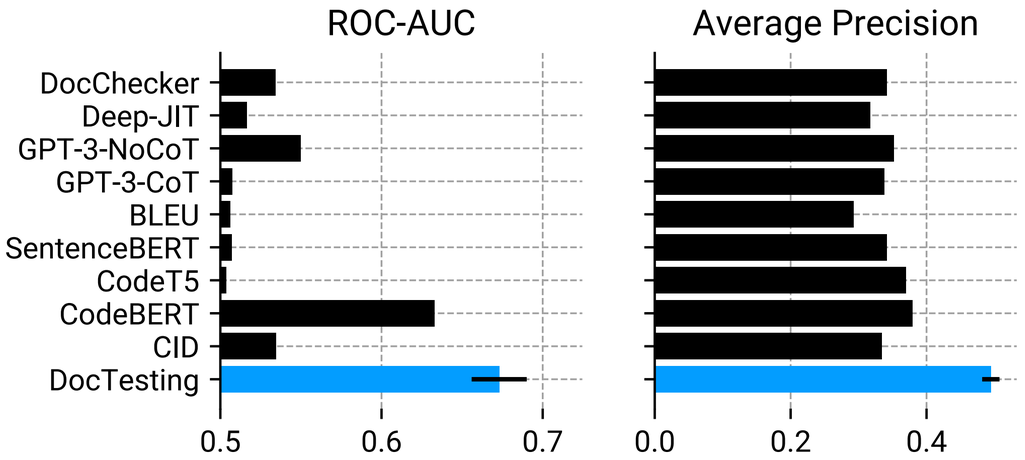
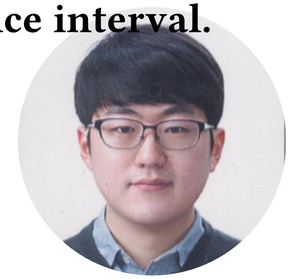


Figure 5: ROC-AUC and AP values compared with baselines. For our approach (blue), we present the mean value from five runs, along with its 95% confidence interval.



Dr. Sungmin Kang
(KAIST)

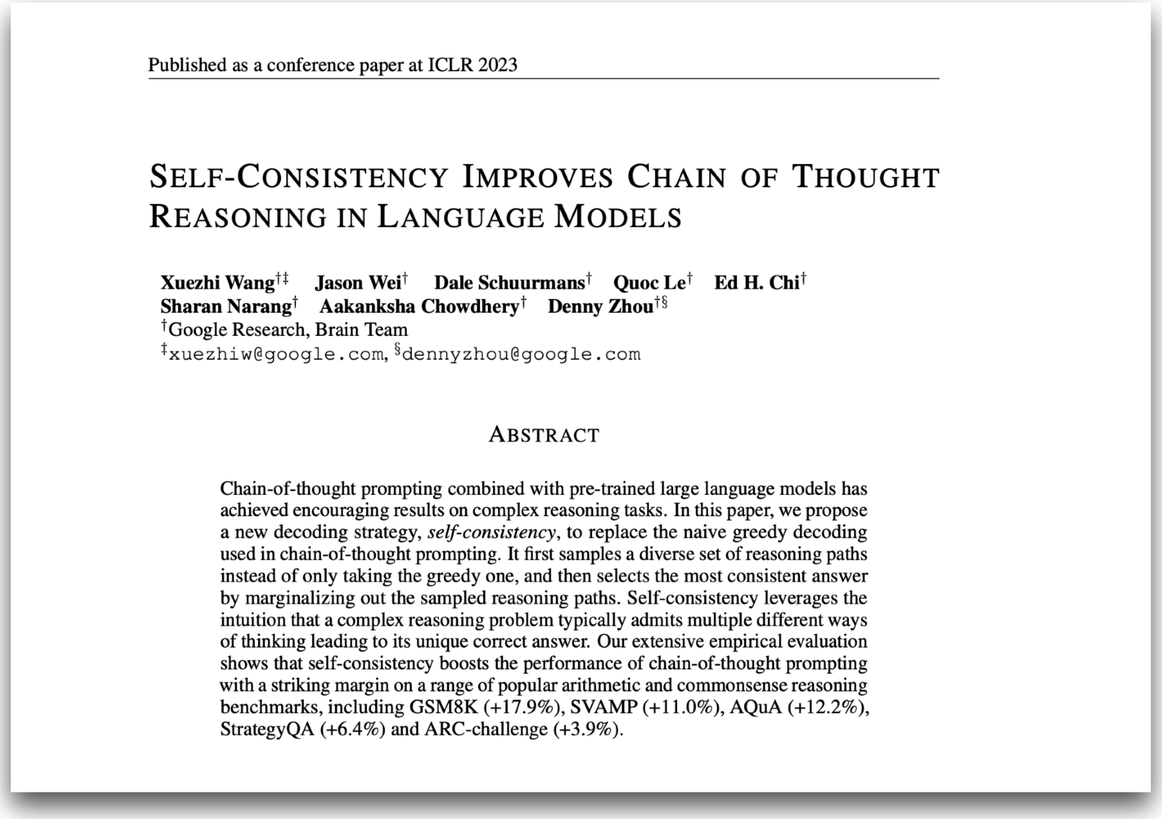
Louis Milliken
(MSc Candidate)

Landscape Analysis

Self-Consistency

Wang et al., ICLR 2023

- When sampling answers from an LLM, take multiple answers with high temperature.
- If there is an answer that has the majority among the sampled answers, it is more likely to be the correct one.



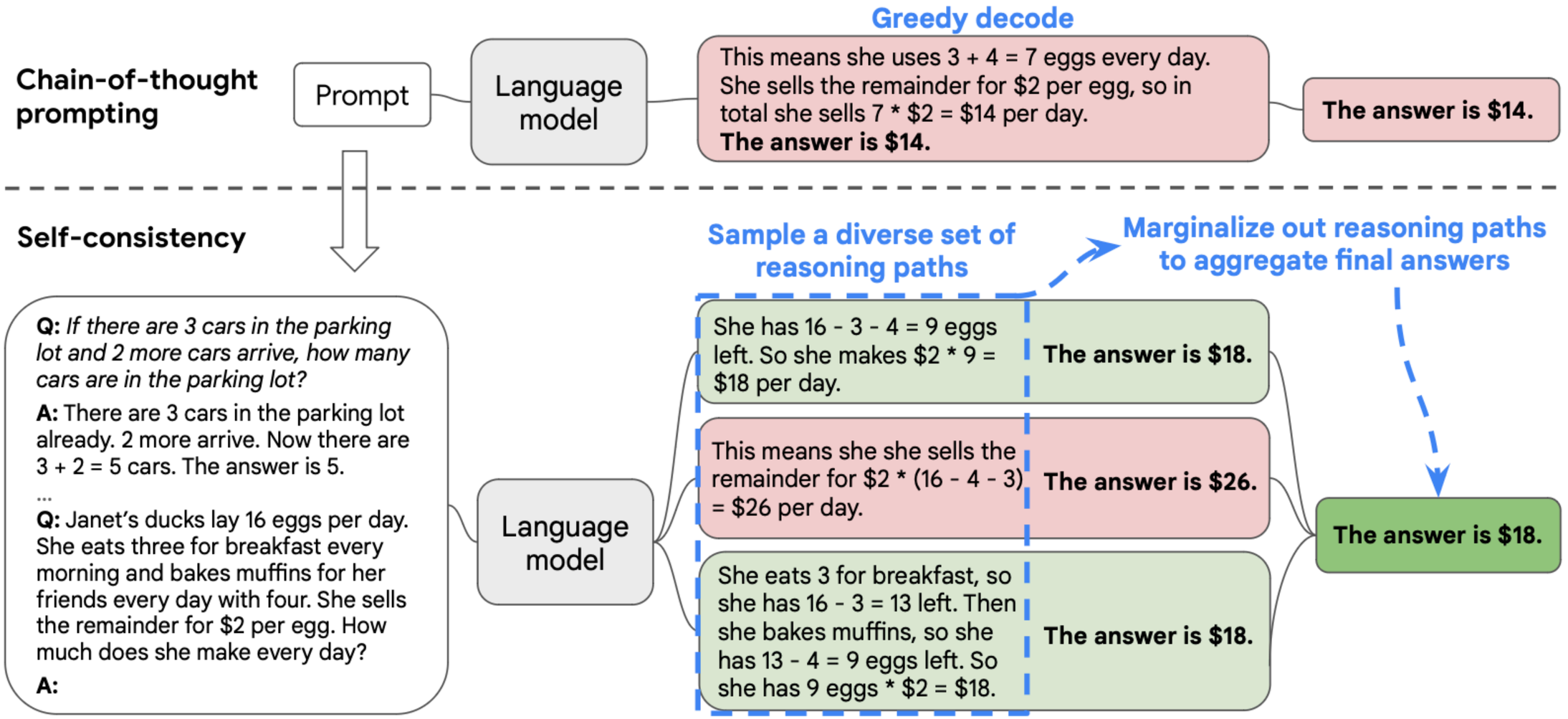
Published as a conference paper at ICLR 2023

SELF-CONSISTENCY IMPROVES CHAIN OF THOUGHT REASONING IN LANGUAGE MODELS

Xuezhi Wang[†‡] **Jason Wei**[†] **Dale Schuurmans**[†] **Quoc Le**[†] **Ed H. Chi**[†]
Sharan Narang[†] **Aakanksha Chowdhery**[†] **Denny Zhou**[†§]
[†]Google Research, Brain Team
[‡]xuezhiw@google.com, [§]dennyzhou@google.com

ABSTRACT

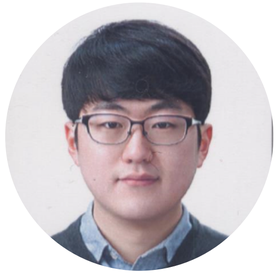
Chain-of-thought prompting combined with pre-trained large language models has achieved encouraging results on complex reasoning tasks. In this paper, we propose a new decoding strategy, *self-consistency*, to replace the naive greedy decoding used in chain-of-thought prompting. It first samples a diverse set of reasoning paths instead of only taking the greedy one, and then selects the most consistent answer by marginalizing out the sampled reasoning paths. Self-consistency leverages the intuition that a complex reasoning problem typically admits multiple different ways of thinking leading to its unique correct answer. Our extensive empirical evaluation shows that self-consistency boosts the performance of chain-of-thought prompting with a striking margin on a range of popular arithmetic and commonsense reasoning benchmarks, including GSM8K (+17.9%), SVAMP (+11.0%), AQuA (+12.2%), StrategyQA (+6.4%) and ARC-challenge (+3.9%).



Wang et al., ICLR 2023

LLM-Based Bug Reproduction

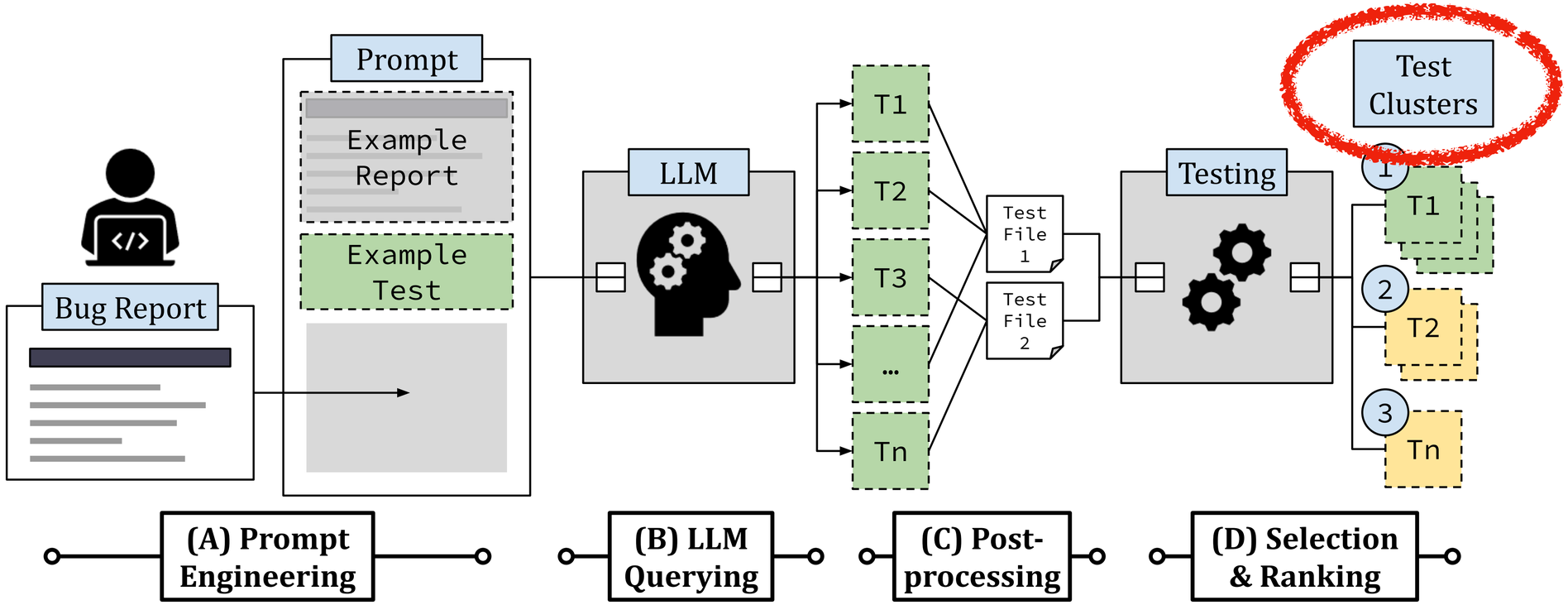
Kang, Yoon & Yoo (ICSE 2023)



Dr. Sungmin Kang
(KAIST)

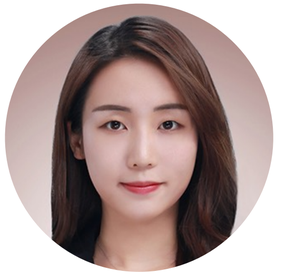
Juyeon Yoon
(PhD Candidate)

Cluster generated tests based on failure messages



LLM-Based Fault Localization

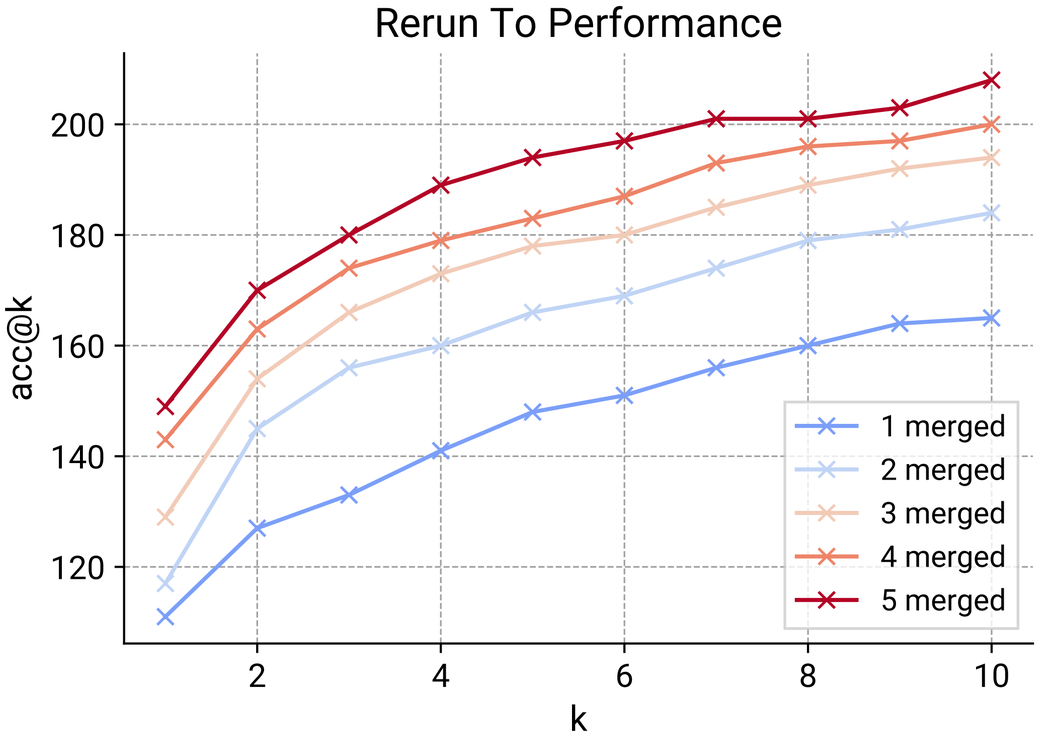
Kang, An & Yoo (FSE 2024)



Dr. Gabin An
(ROKU Korea)

Dr. Sungmin Kang
(KAIST)

Family	Technique	acc@1	acc@3	acc@5
Predicate Switching		42	99	121
Stack Trace		57	108	130
Slicing (frequency)		51	96	119
MBFL	MUSE	73	139	161
	Metallaxis	106	162	191
SBFL	Ochiai	122	192	**218**
	DStar	125	**195**	216
	SBFL-F	34	66	78
LLM-Based	LLM+Test	81	94	97
	AutoFL	**149**	180	194



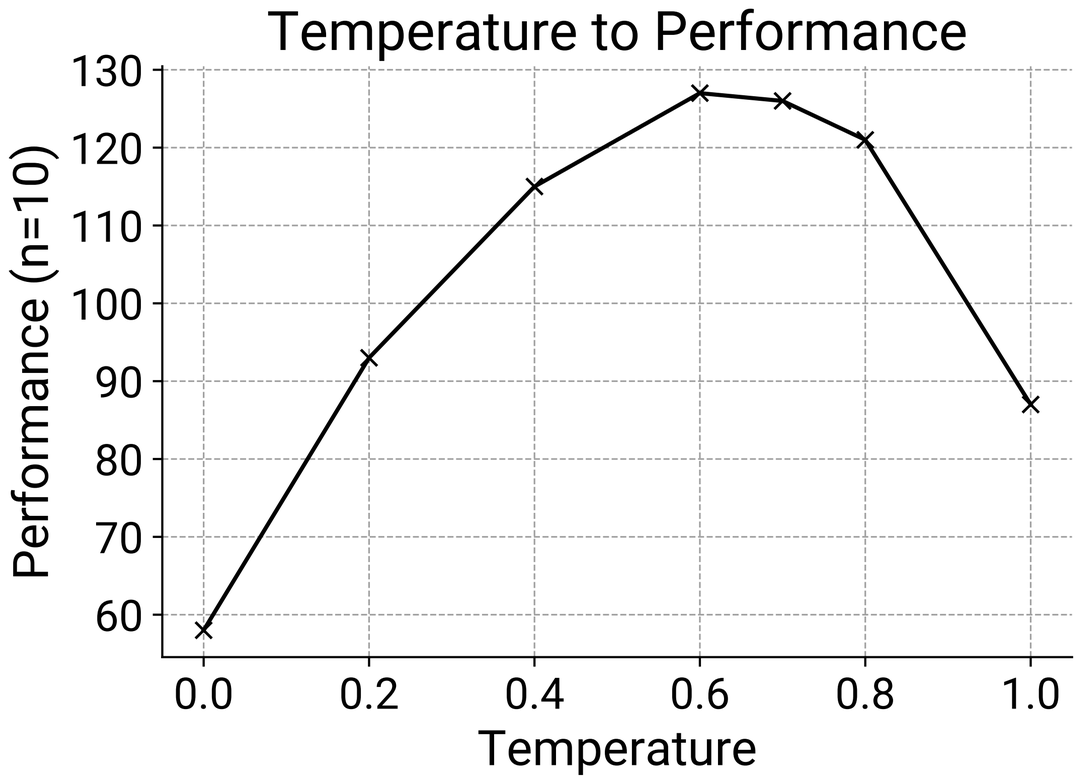
Why does this work?

- Wang et al.’s original intuition: “there are many reasoning paths to the correct solutions, but only one way to arrive at a specific incorrect solution”
- My first reaction: “surely there are infinite ways to arrive at a single incorrect solution!”
- My second reaction: “oh, it is probably assumed that the LLM is at least **trying**… that is, there are infinite total nonsense ways to arrive at a specific incorrect solution, but perhaps *fewer ways to move from the question to a specific incorrect solution while trying to appear plausible*”

Libro Journal Ext.

Kang et al., Under Review

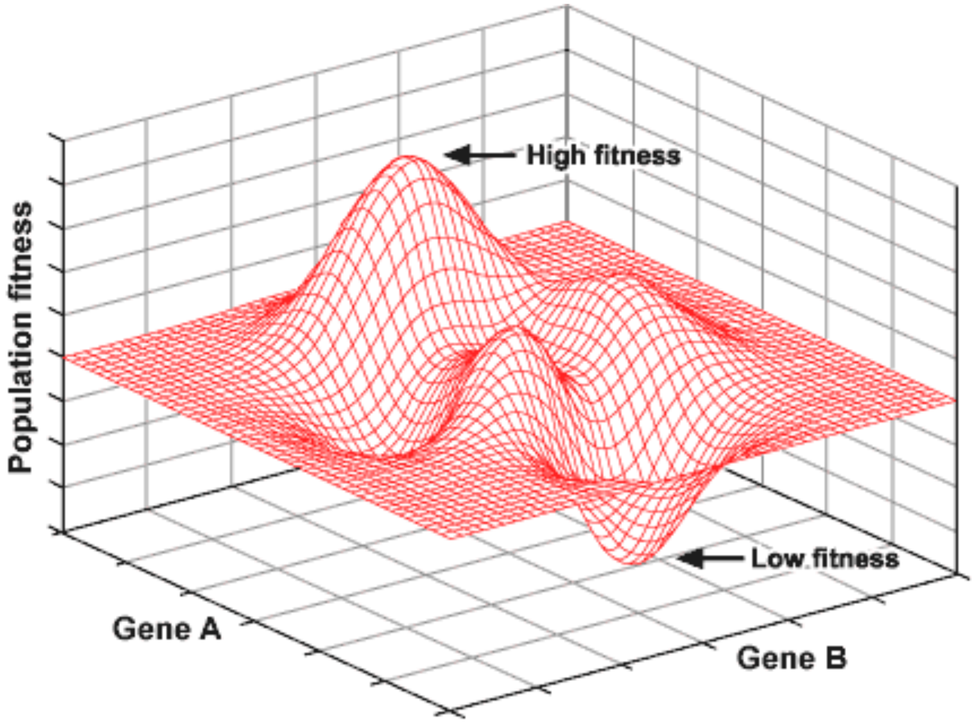
- Empirical evidence for my second reaction…?
- Too high a temperature —> too random sequence sampling —> not really trying to make sense —> self-consistency seems to break down…



Landscape Analysis

Fitness Landscape Analysis from Optimization Literature

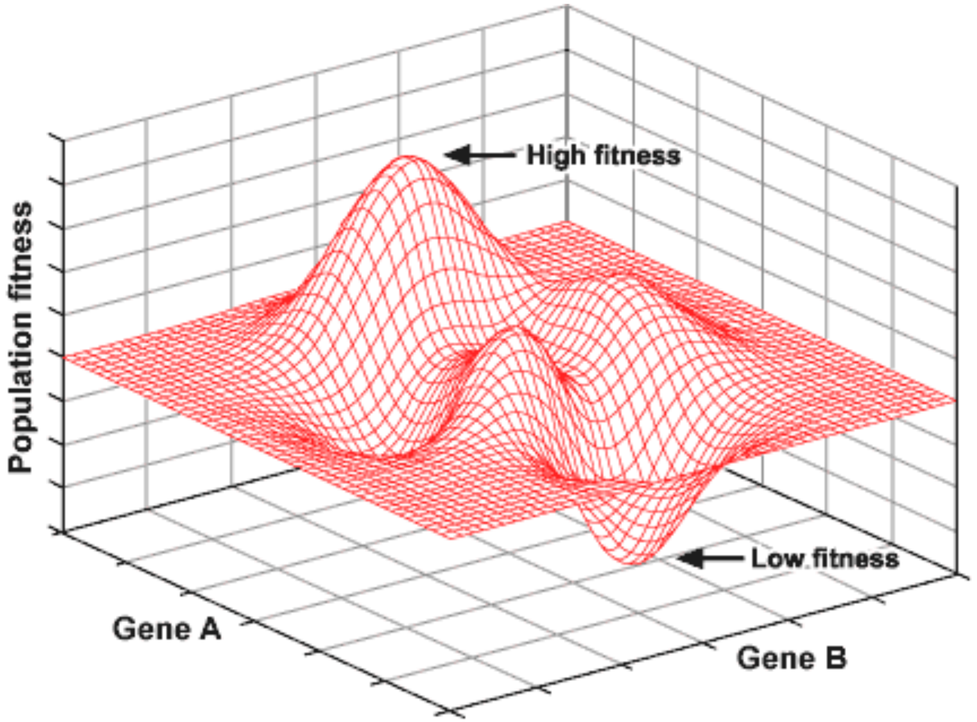
- Fitness Landscape = [solution space] × [fitness dimension]
- Optimization is essentially climbing up hills to get higher fitness
- What if we see LLM-based solution generation as an optimization process?
  - What would be the landscape that results in self-consistency?



Landscape Analysis

Fitness Landscape Analysis from Optimization Literature

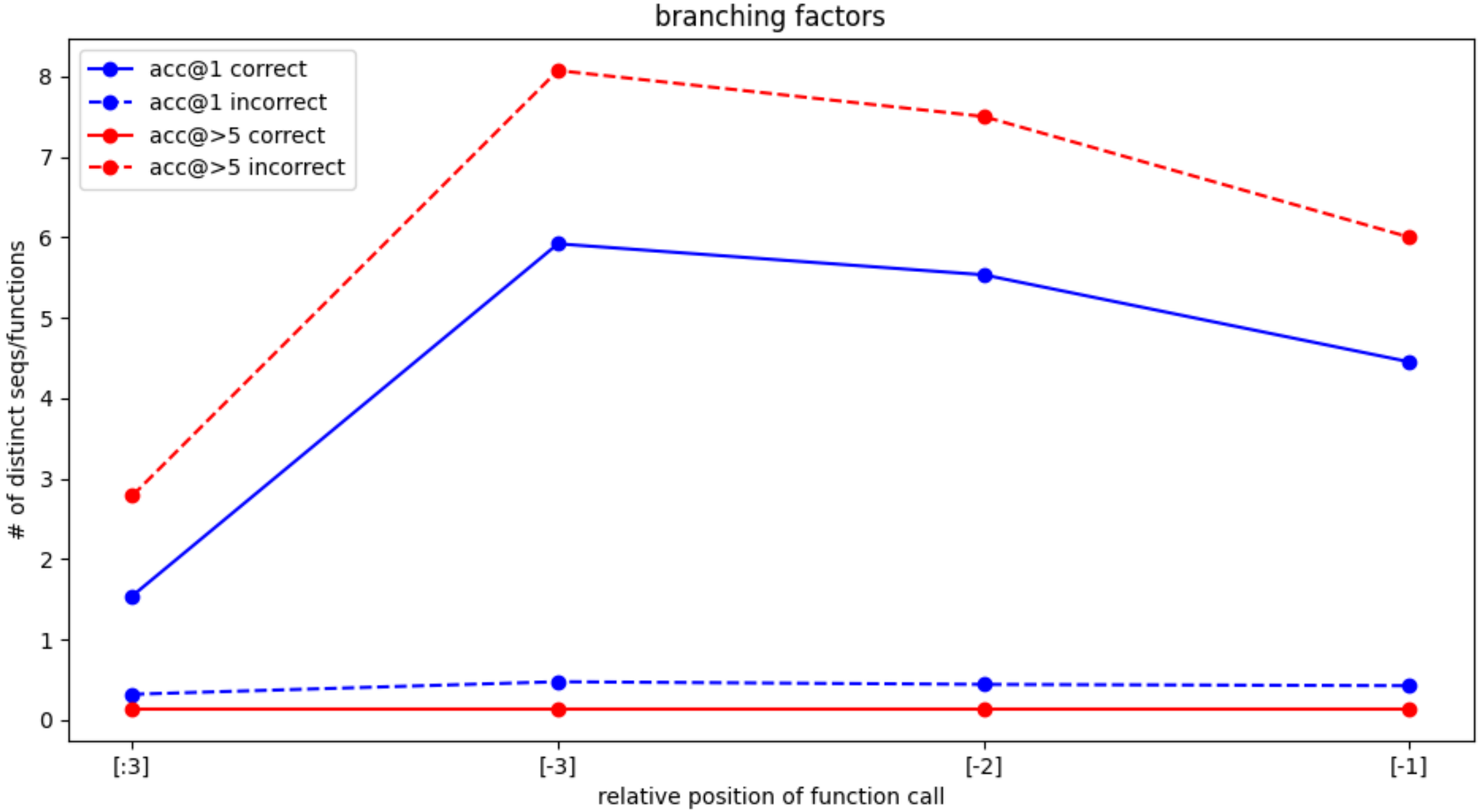
- With problems for which the self-consistency works, I hypothesize that:
  - The tallest hill is also the largest; there are multiple starting points and pathways to the top
  - Smaller hills (=incorrect solutions) have smaller base area, resulting in fewer pathways to their top

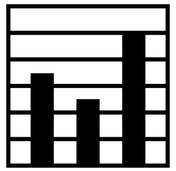


Analyzing AutoFL Function Calls

(a VERY preliminary analysis, perhaps “I” am wrong not →)

Naryeong Kim
(Intern)





Mathematical Modelling

Performance Prediction

Dr. Sungmin Kang
(KAIST)

- “So taking more samples increases the performance, but it also costs more. Can I predict the performance of LLM based on the current sample size?”
- ~~We~~ Dr. Kang has applied Laplace estimator to LLM performance prediction of LIBRO and AutoFL: it estimates the probability of an event that has never happened
- Let N be the total number of problems we want to solve, $S(x)$ the number of successful answers when sampling x times. This leaves $N - S(x)$ unsolved problems.

Laplace Estimator

Dr. Sungmin Kang
(KAIST)

- Originally, given that c_s is the number of successful attempts, and α a hyper parameter, it is defined as:
  - $Lap(x, c_s) = \frac{c_s + \alpha}{x + 2\alpha}$, which becomes $\frac{1}{2}$ when $x = 0,\ c_s = 0$
- Using $p = \frac{S(1)}{N}$, we redefine this as: $Lap'(x, c_s) = \frac{c_s + 2p}{x + 2}$
- What can we do with this?

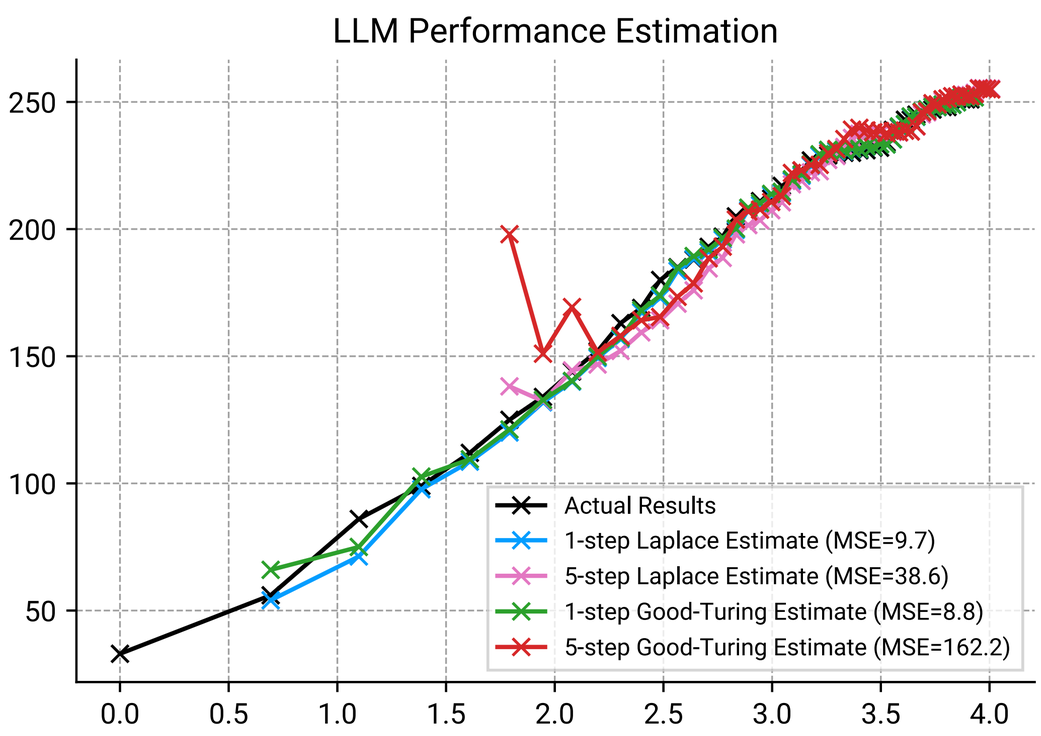
Laplace Estimator

Dr. Sungmin Kang
(KAIST)

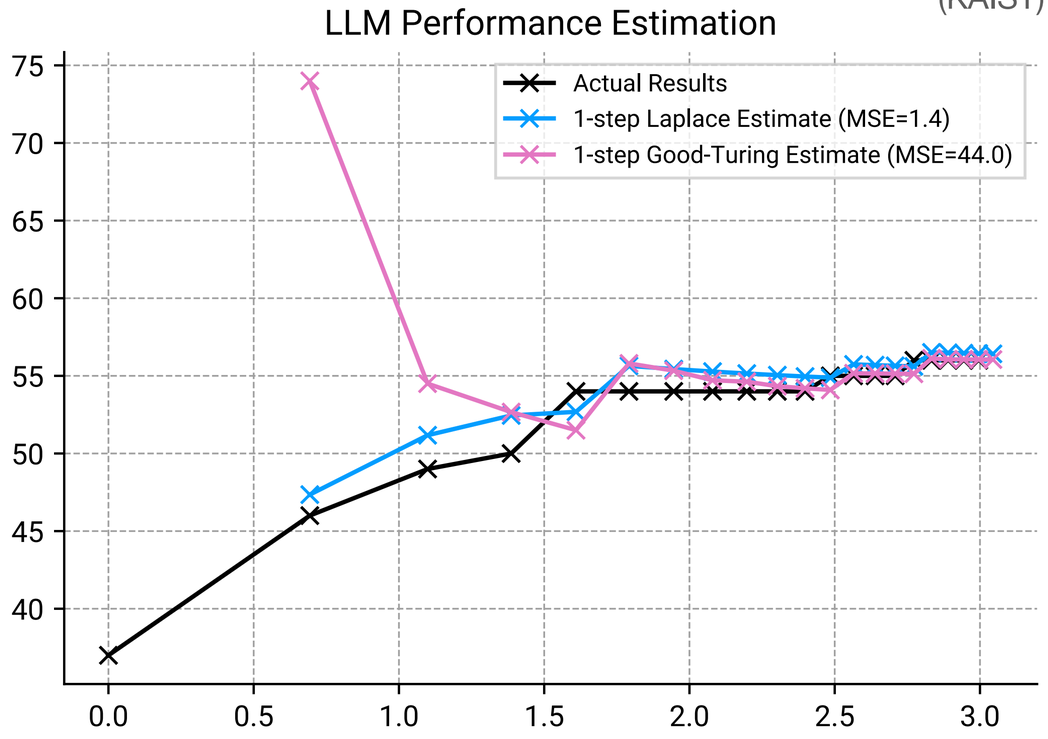
- Suppose we are trying to solve N problems, out of which $S(x)$ were solved using x samples (=attempts). How many successful attempts do I get, if I take x' more samples ($x' > x$) ?
  - Remaining problems: $N - S(x)$
  - Per remaining problems, additional attempts: $x' - x$
  - For these attempts, the probability of success: $Lap'(x,0)$
  - $\therefore E[S(x')] - S(x) = Lap'(x,0) \times (N - S(x)) \times (x' - x)$

Laplace Estimator

Dr. Sungmin Kang
(KAIST)



(a) LIBRO

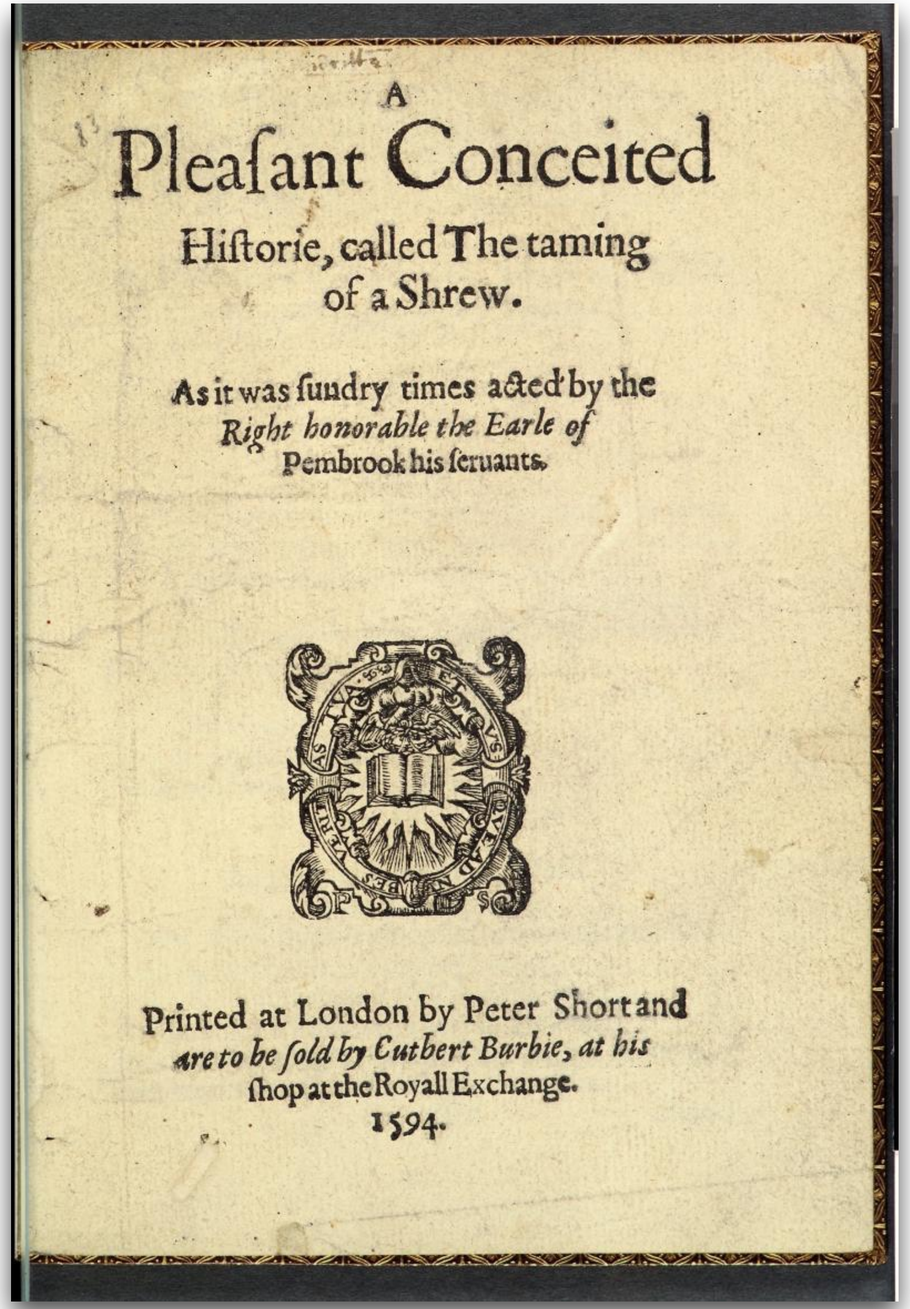


(b) AutoFL [KAY23]

Figure 4: Performance Prediction. Lower Mean-squared error (MSE) is better.

Conclusion

- Healthy skepticism about LLMs required (personal belief).
- *“We achieved metric X value Y with the biggest LLM so far”* gets boring.
- Still too big to treat as a white box, but I believe there are analyses and modeling we can do.



A
Pleaſant Conceited
Hiſtorie, called The taming
of a Shrew.

As it was ſundry times acted by the
Right honorable the Earle of
Pembrook his ſeruants.

Printed at London by Peter Short and
are to be ſold by Cutbert Burbie, at his
ſhop at the Royall Exchange.
1594.

William Shakespeare, aka, the Bard

Taming of a shrew

(말괄량이 길들이기)