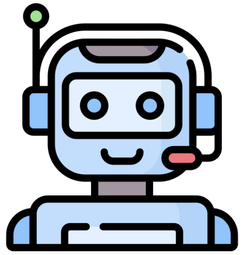
Solving Real-World Tasks with AI Agents

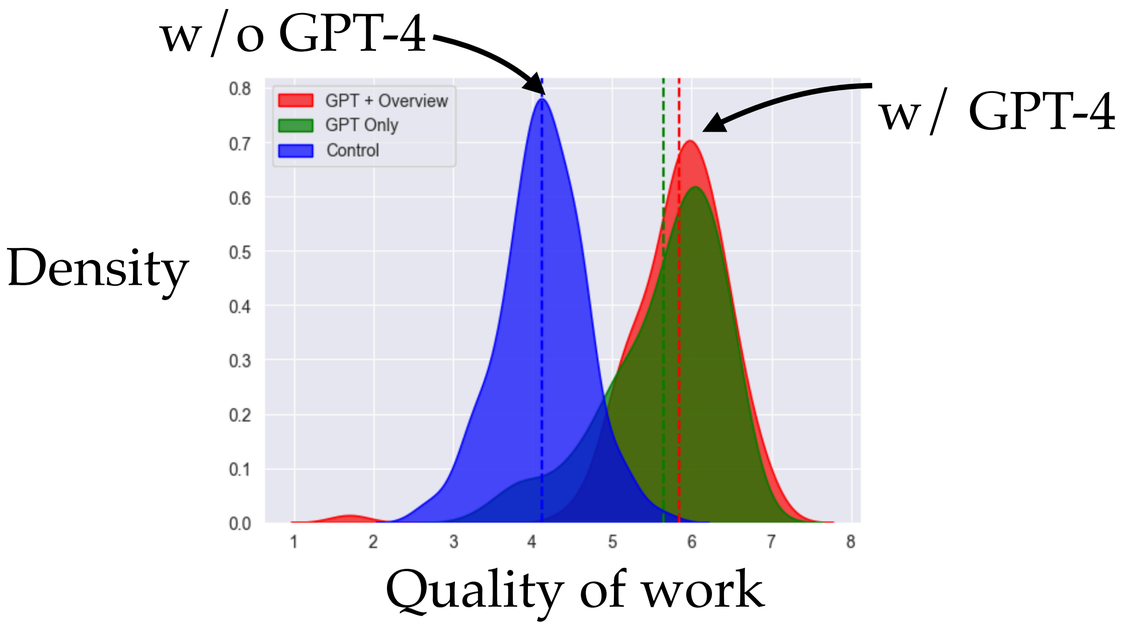


Shuyan Zhou
Language Technologies Institute
Carnegie Mellon University
shuyanzh@cs.cmu.edu
shuyanzhou.com

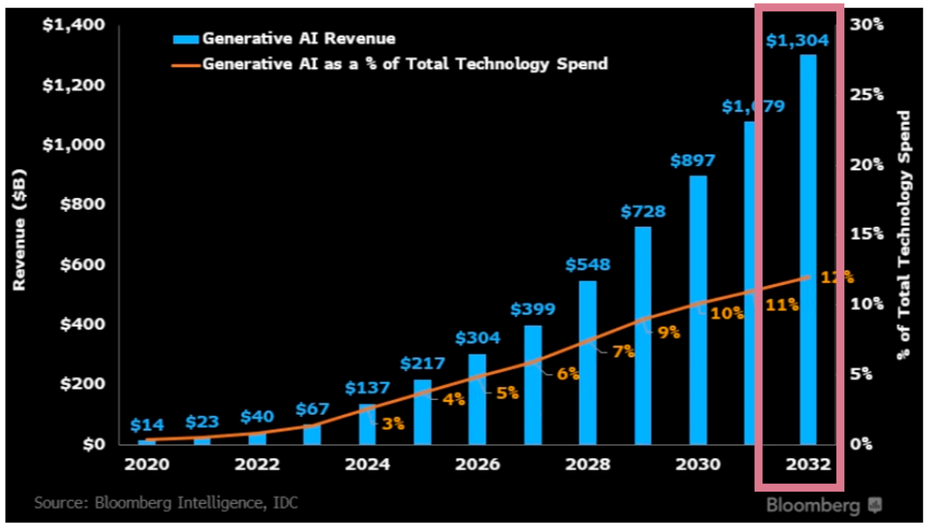
Carnegie Mellon University
Language Technologies Institute

Carnegie Mellon University

LLMs are useful, people are optimistic about the future



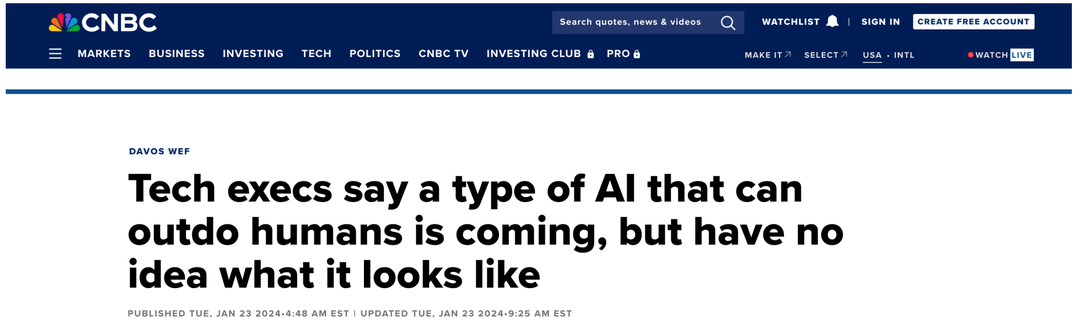
$1.3T revenue from generative AI in 2032



Sparks of Artificial General Intelligence: Early experiments with GPT-4

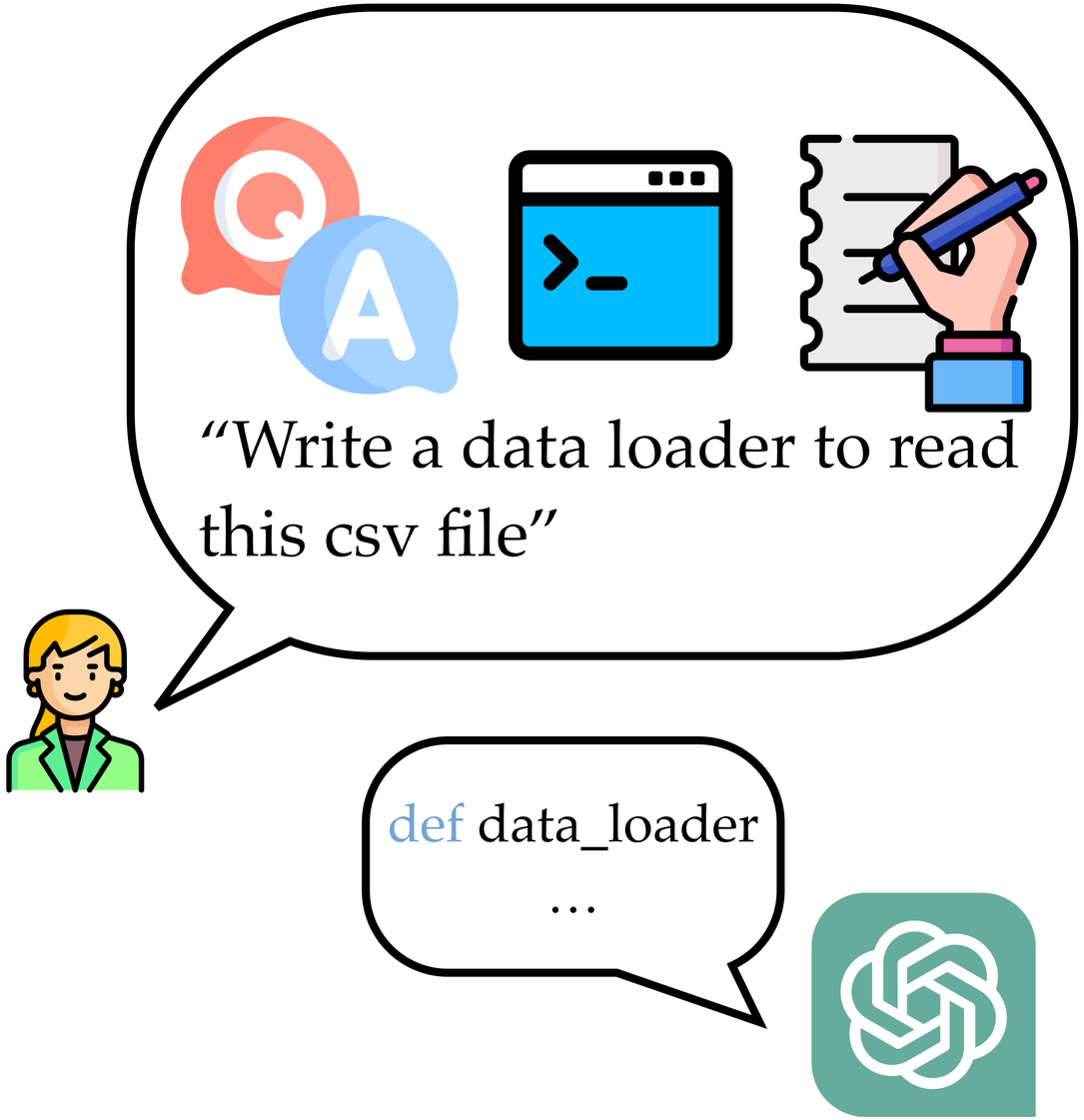
Sébastien Bubeck Varun Chandrasekaran Ronen Eldan Johannes Gehrke
Eric Horvitz Ece Kamar Peter Lee Yin Tat Lee Yuanzhi Li Scott Lundberg
Harsha Nori Hamid Palangi Marco Tulio Ribeiro Yi Zhang

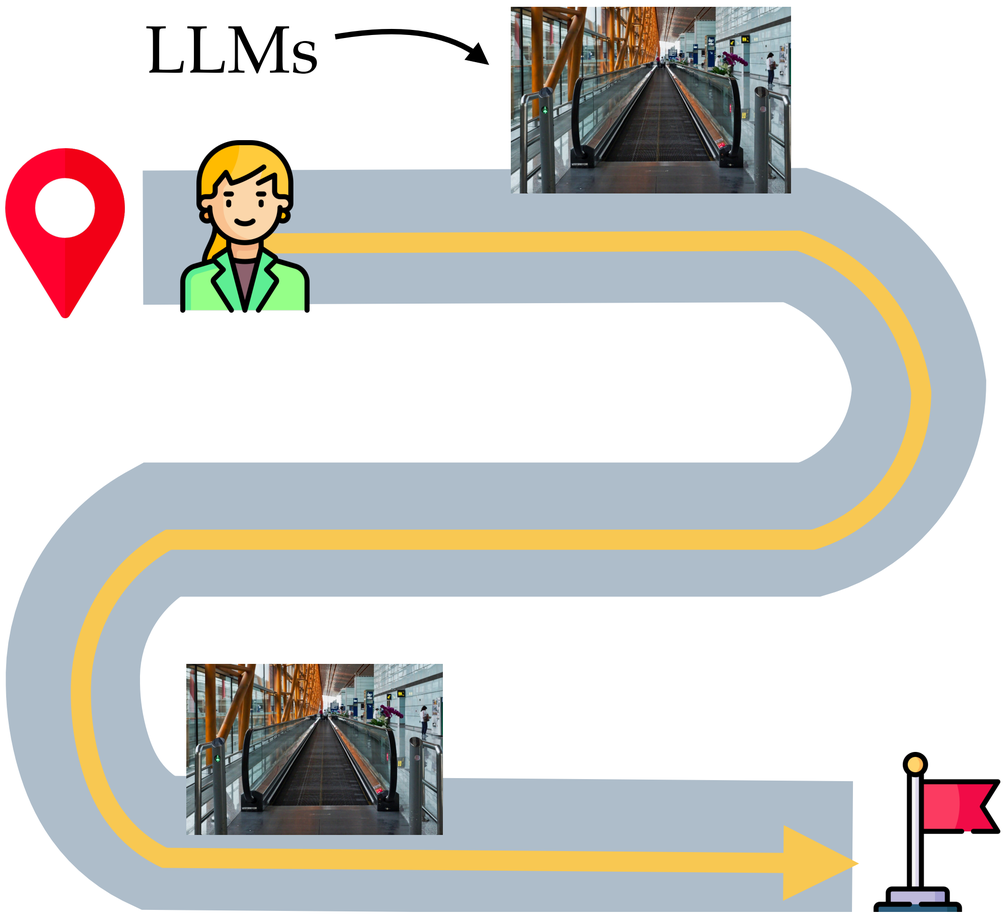
Microsoft Research



[Dell'Acqua et al, 2023] [Bloomberg 2023]

LLMs can assist humans in many self-contained tasks

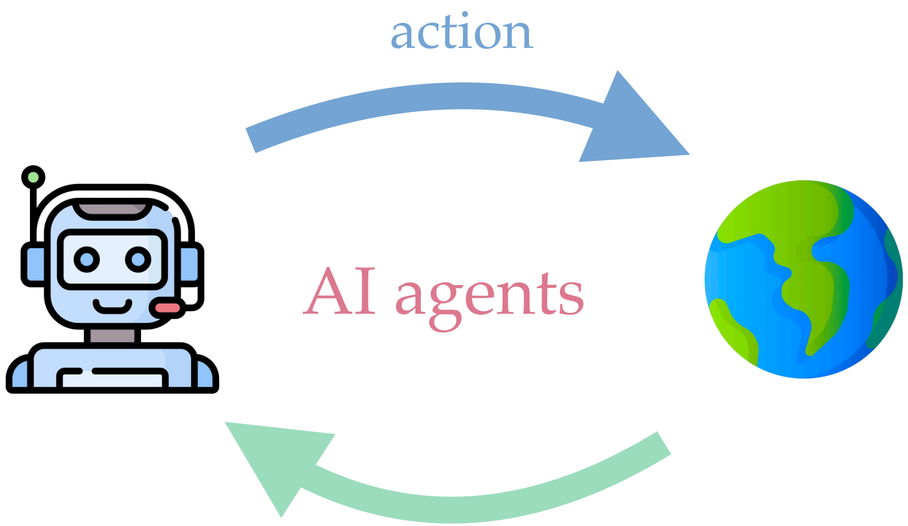




Speed up a small part of a task

Not automate the tasks in an end-to-end fashion

The dream of AI is far more wild

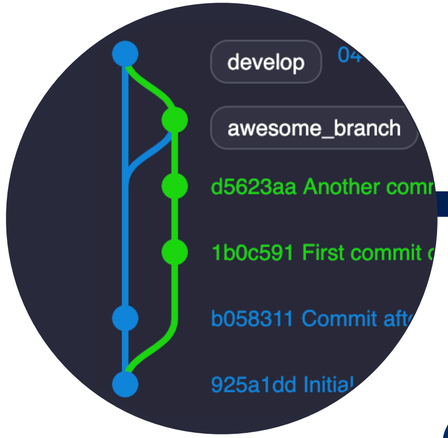


My research goal

Automate *various* tasks with minimal human intervention

Perform scientific research

Develop software



Reproduce results

Experiments

Literature review

Personalized health and wellness

Finance and growth management

Questions to answer

How good are strong LLMs (e.g., GPT-4)? How can we perform reliable evaluation?

What are the fundamental gaps between LLMs and AI agents?

How could we mitigate the gaps?

Talk Overview

How good are LLMs?

Natural language has inherent limitations

LLMs know up to a cutoff date

Evaluating AI agents

Speaking AI's "language"

Learning new knowledge by reading

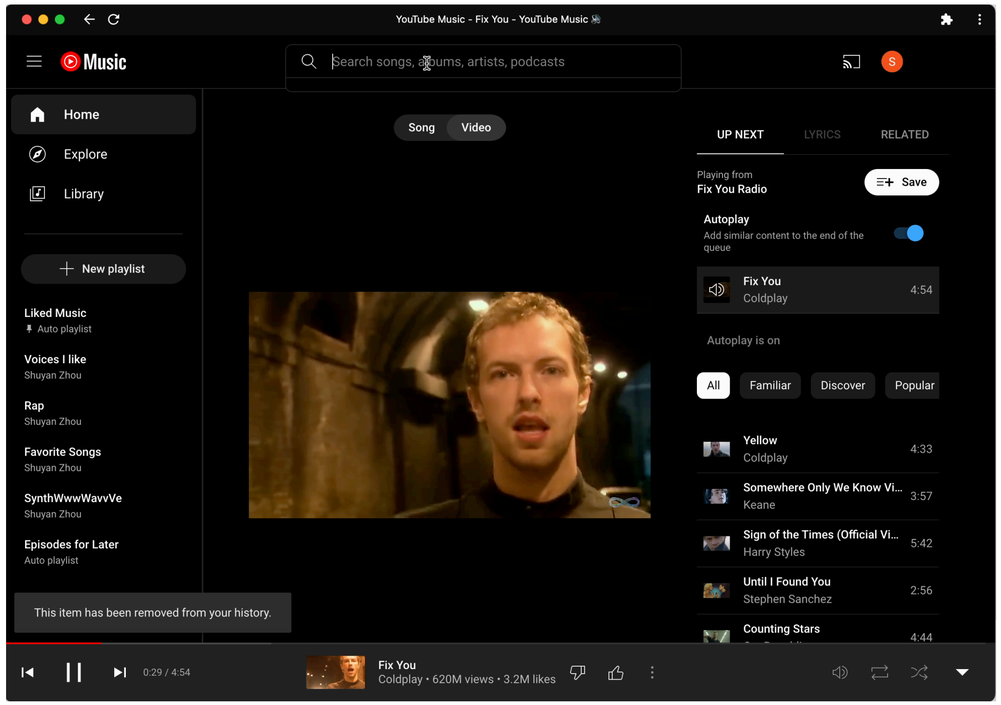
- ***Zhou*** et al., WebArena, ICLR 2024*
- *Wang, Cuenca, **Zhou** et al., MCoNaLa, F-EACL 2023*
- *Wang, **Zhou** et al., ODEX, F-EMNLP 2023*

- ***Zhou** et al., PaP, SUKI 2022*
- ***Zhou*** et al., PaL, ICML 2023*
- *Madaan, **Zhou** et al., CoCoGen, EMNLP 2022*
- *Zhang, Xu, Yang, **Zhou** et al, Crepe, F-EACL 2023*

- ***Zhou** et al., DocPrompting, ICLR 2023*
- ***Zhou*** et al., Hierarchical Procedural KB, ACL 2022*

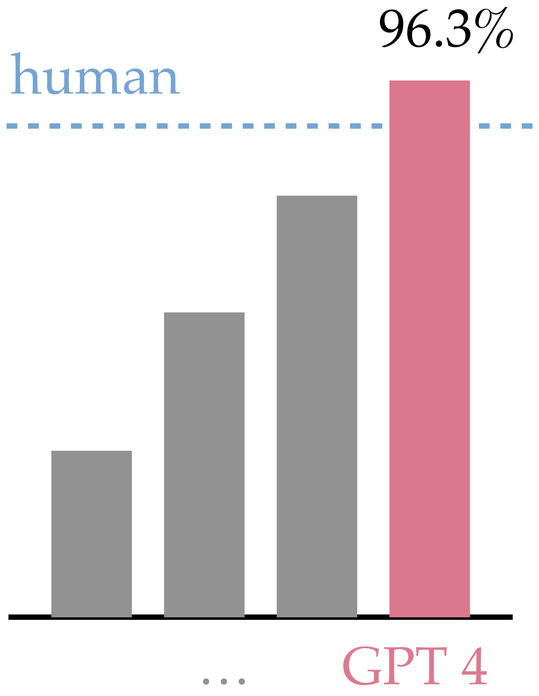
Significant gap in benchmarks vs real-world applications

[Liu et al., Miniwob++, 2018]



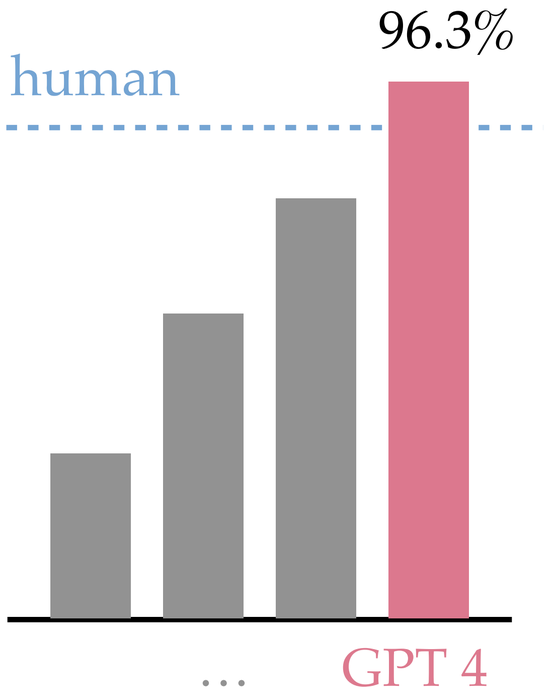
"Play my favorite music"

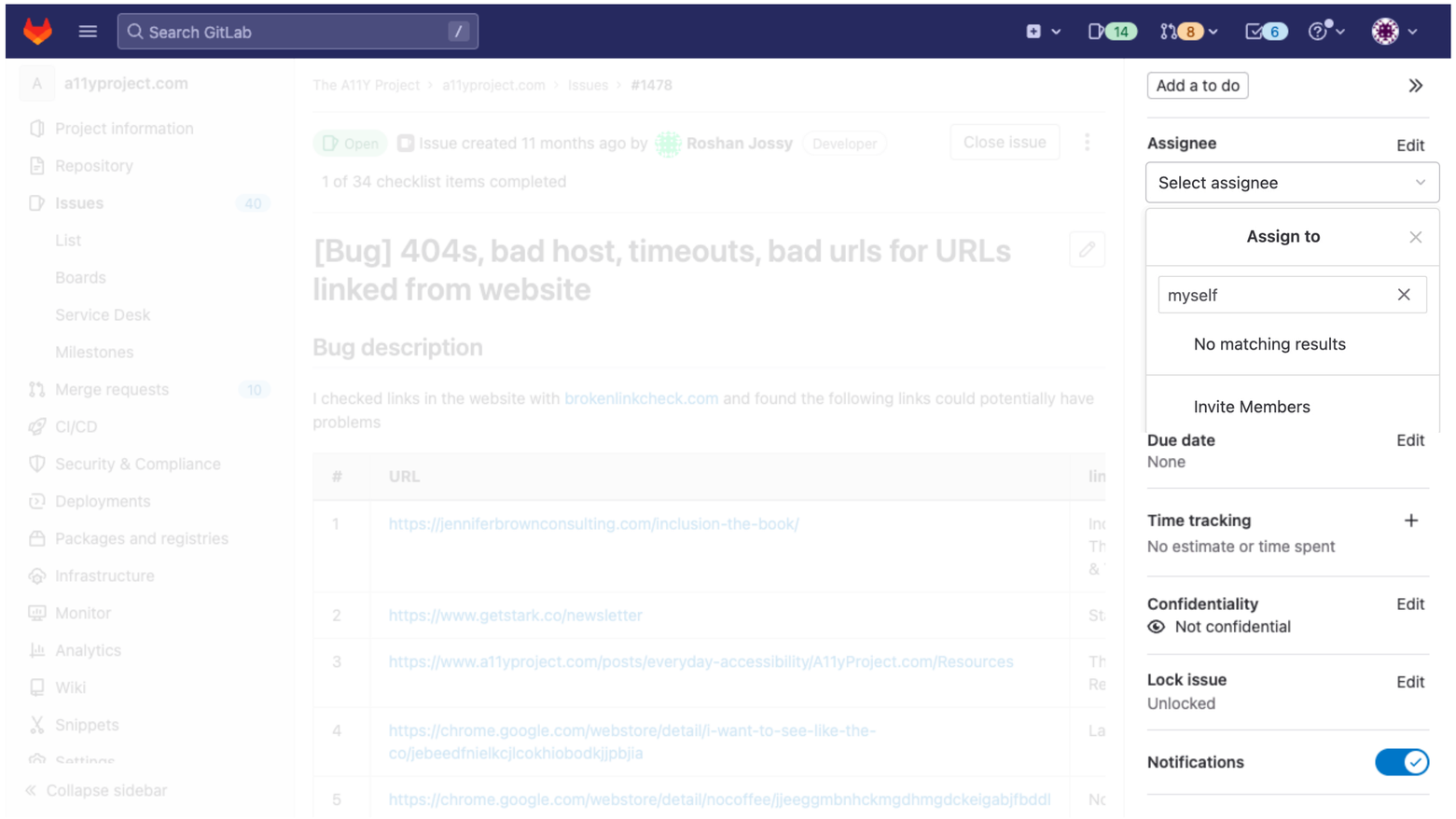
Task-solving rate on Miniwob++



Significant gap in benchmarks vs real-world applications

Task-solving rate on Miniwob++



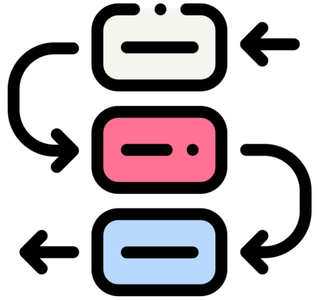


"Assign this issue to myself"

Requirements for the agent evaluation

Realistic interactive environment

Useful & complex tasks



Existing evaluations make trade-offs between them

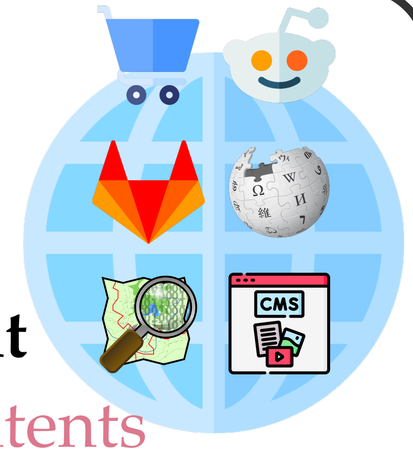
Reliable evaluation

Easy extendability

Zhou et al, WebArena: A realistic web environment for building autonomous agents, ICLR 2024*

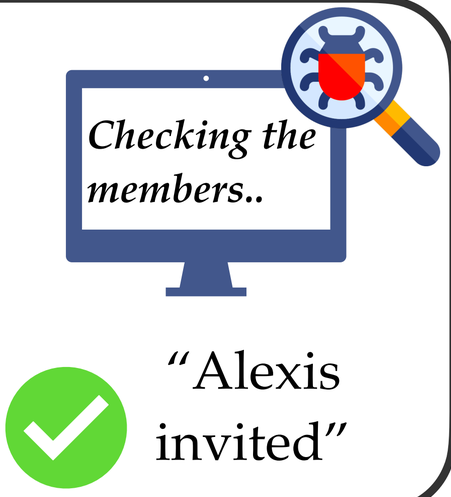
WebArena fulfills all requirements without compromise

Realistic interactive environment
with rich contents



Useful & complex tasks

Reliable evaluation



Easy extendability

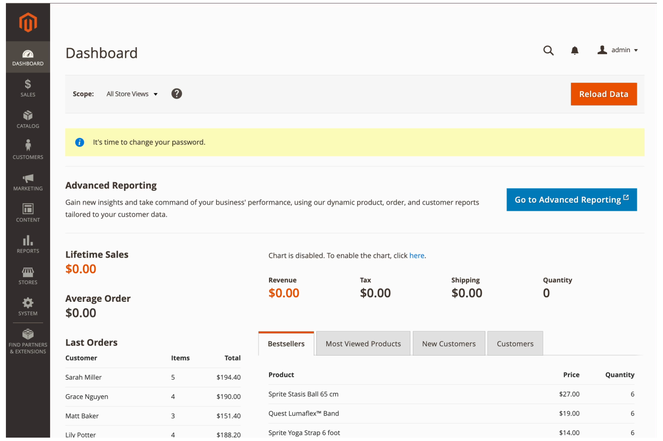
Zhou et al, WebArena: A realistic web environment for building autonomous agents, ICLR 2024*

Example task in WebArena

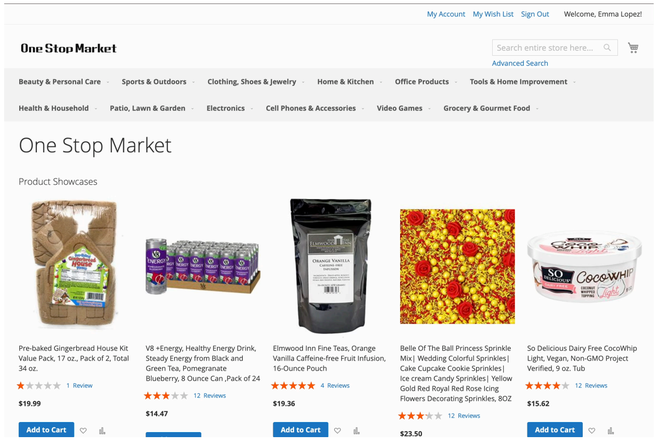
Shop owner 

Find the customer who h[illegible]re over the past 56 days.
Send the customer some [illegible]

Customer appreciation task



Identify the customer by examining the order history in the store portal



Buy some flowers online to the customer

Outcome-based evaluation

- A new order with flowers

Order # 000000190

Product Name

ShineBear Eternal Flowers Dried Flowe
Fresh Flower Live Rose Enchanted Glas
Box - (Colo[illegible]wer Glass)
Color
Blue / Flower Glass

flowers

- Shipped to Alex Martin

Order Information

Shipping Address

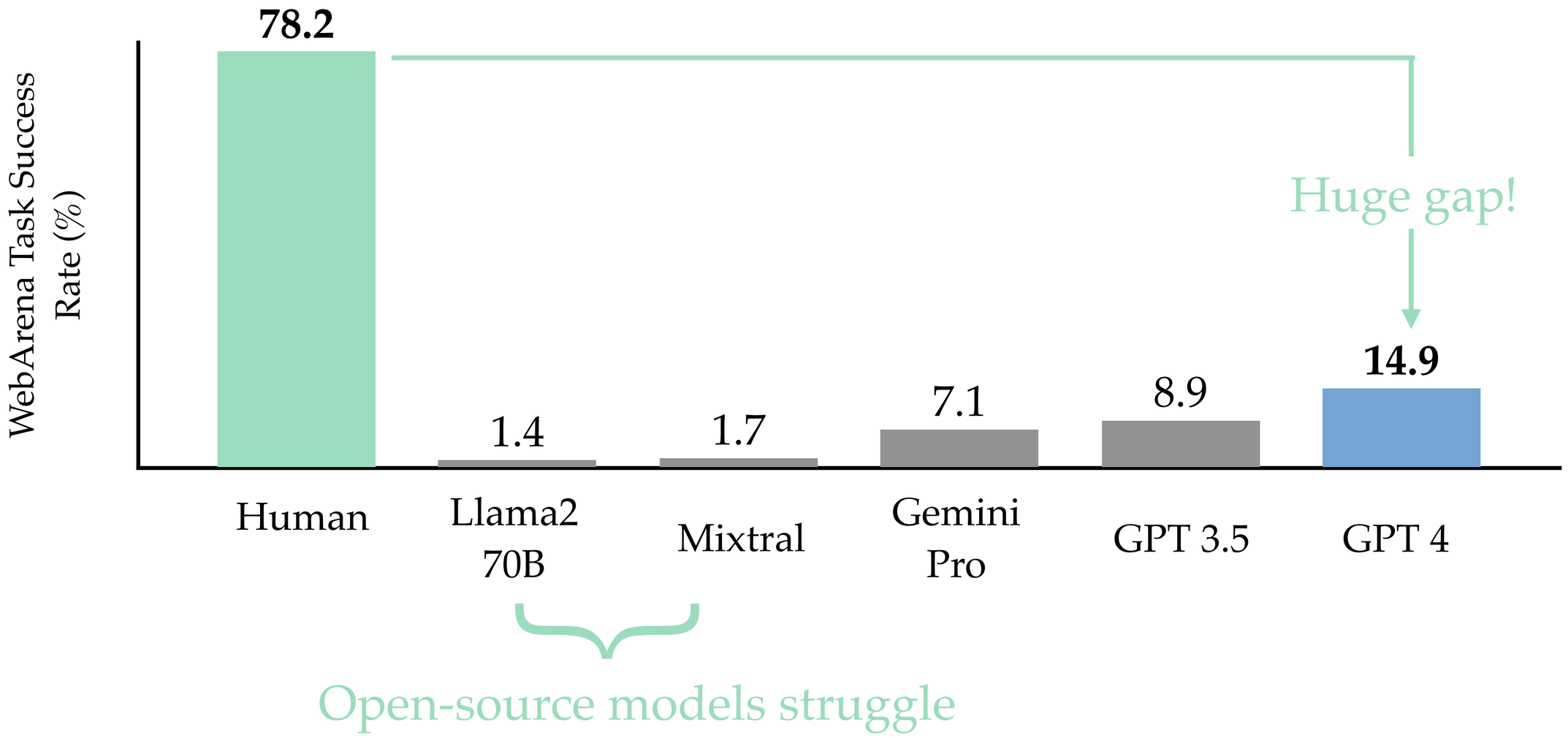
Alex Martin
123 Main Street
New York, New York, 10001
United States
T: 2125551212

812 long-horizon, realistic computer tasks

Zhou et al, WebArena: A realistic web environment for building autonomous agents, ICLR 2024*

LLMs are the critical yet early step toward AI autonomy

LLMs lack several critical capabilities to be AI agents



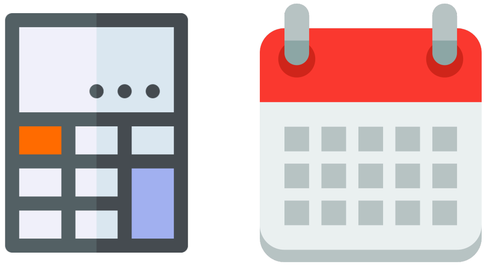
Zhou et al, WebArena: A realistic web environment for building autonomous agents, ICLR 2024*

LLMs lack critical capabilities to be AI agents

Tool use

Alex's total spend is
78.56 x 7 + 46.7 = 543.6

56 days ago is 5/20/2023



AI agents

- Employ tools to enhance accuracy and expand capabilities

LLMs

- Scarce in natural language corpus
- Not consider tool use in standard LLM development

Zhou et al, WebArena: A realistic web environment for building autonomous agents, ICLR 2024*

LLMs lack critical capabilities to be AI agents

Abstract reasoning

Fork `metaseq`

Fork `transformers`
Fork all repos owned by Meta

AI agents

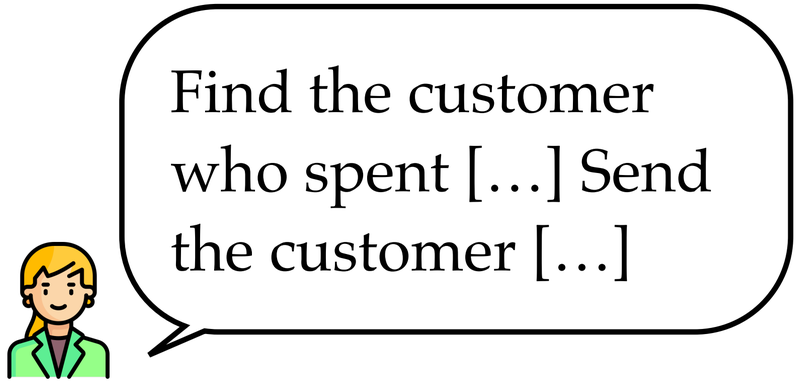
- Learn the common principles
- Maintain steady and reliable performance

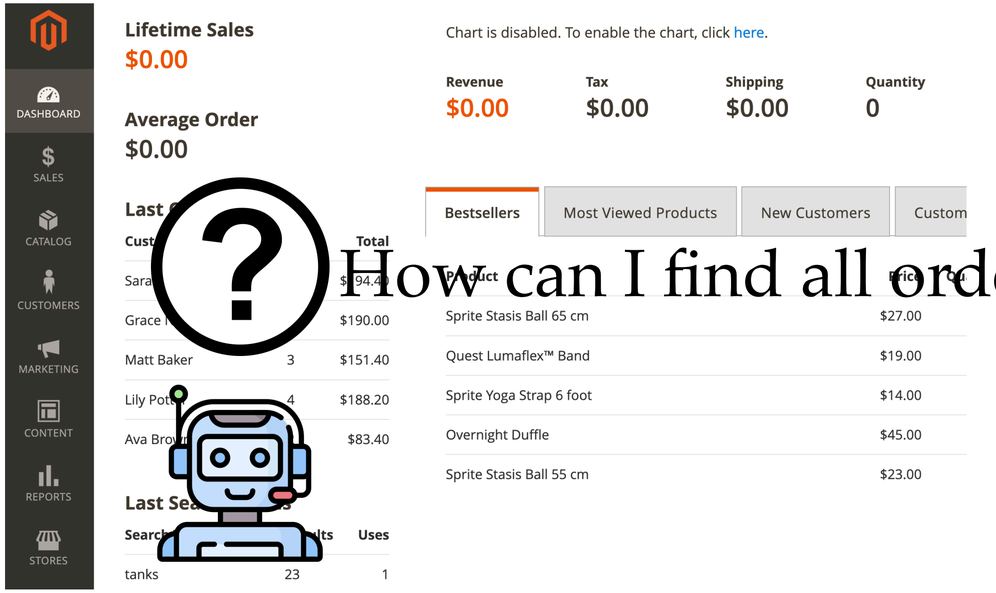
LLMs

- Inconsistent performance across conceptually similar tasks

Zhou et al, WebArena: A realistic web environment for building autonomous agents, ICLR 2024*

LLMs lack critical capabilities to be AI agents



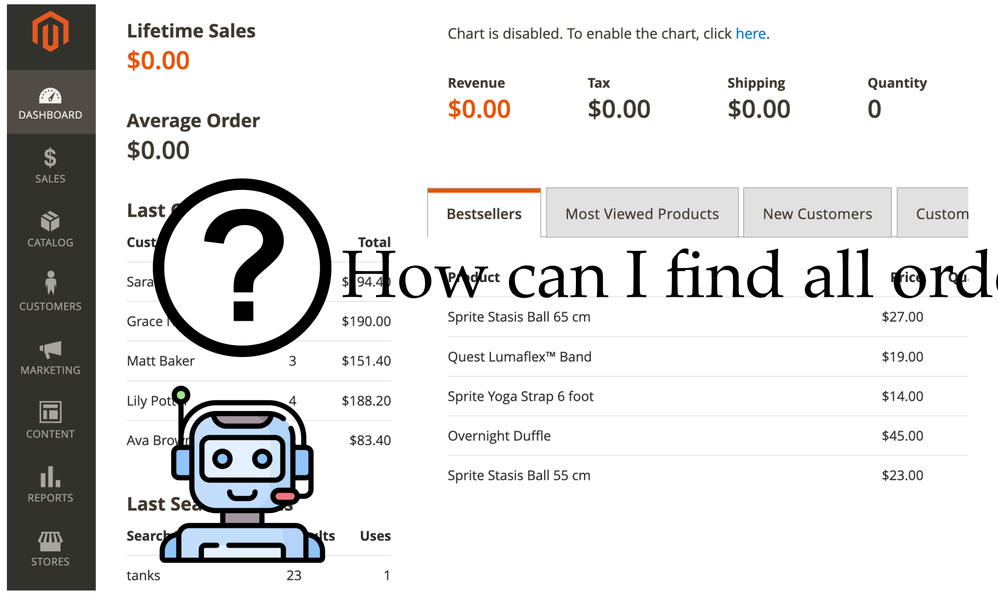


How can I find all orders?

Zhou et al, WebArena: A realistic web environment for building autonomous agents, ICLR 2024*

LLMs lack critical capabilities to be AI agents

Up-to-date knowledge



How can I find all orders?

GPT-4 knowledge cutoff: Sep 2021

WebArena application version: Jan 2023

AI agents

- Up-to-date knowledge to deal with the evolving world

LLMs

- Knowledge of LLMs is limited by the training cutoff

Zhou et al, WebArena: A realistic web environment for building autonomous agents, ICLR 2024*

Tool use

Abstract reasoning

Speaking AI's "language"

Up-to-date knowledge

Learning by reading docs

Generating natural language for various tasks

Alex Martin made three orders: $47.51 on 9/18/2023, $765.8 on 1/1/2024 and $35.4 on 1/9/2024. How much he spent in my store in the last 56 days?

Today is 1/20/2024. I first subtract 20 days […] The date 56 days ago is 12/20/2023
[…] Order 1 was placed on 9/18/2023, which is not within the last 56 days
[…] 765.8 + 35.4 = $785.4

[Wei et al., Chain-of-thought]

Natural language exhibits limitations in performing tasks

Today is 1/20/2024, Alex made three orders: $47.51 on 9/18/2023, $765.8 on 1/1/2024, $35.4 on 1/9/2024. How much has he spent in the last 56 days?

Today is 1/20/2024. I first subtract 20 days […] The date 56 days ago is 12/20/2024 ❌
[…] Order 1 was placed on 9/18/2023, which is not within the last 56 days
[…] 765.8 + 35.4 = $785.4 ❌

[Wei et al., Chain-of-thought]

⚠️ **Confine reasoning and solving within LLMs**

***Zhou** et al, Procedures as programs: hierarchical control of situated agents through natural language, SUKI 2022*

Natural language exhibits limitations in performing tasks

Today is 2/13/2024, Alex made three orders: $47.51 on 9/18/2023, $765.8 on 1/1/2024, $35.4 on 1/9/2024. How much has he spent in the last 192 days

Today is 1/20/2024. I first subtract 20 days […] The date 56 days ago is 12/20/2024
[…] Order 1 was placed on 9/18/2023, which is not within the last 56 days
[…] 765.8 + 35.4 = $785.4

?

Today is 2/13/2024. I first subtract 13 days […] The date 192 days ago is 8/5/2023.
[…] Order 1 was placed on 9/18/2023, which is within the last 192 days
[…] 47.51 + 765.8 + 35.4 …

[Wei et al., Chain-of-thought]

Confine reasoning and solving within LLMs

Express solutions at the example level

Zhou et al, Procedures as programs: hierarchical control of situated agents through natural language, SUKI 2022

Maybe AI agents should speak another "language", but what is that?

Solving various tasks by reasoning with programs (PaL)

Today is 1/20/2024, Alex made three orders: $47.51 on 9/18/2023, $765.8 on 1/1/2024, $35.4 on 1/9/2024. How much has he spent in the last 56 days?

```
[...]
The first order is $47.51
It was made on 9/18/2023
[...]
Now check if the first order
was placed within the period
9/18/2023 is before the period,
so it is not included
[...]
```

```
So the answer is $801.2
```

[Wei et al., Chain-of-thought]

```
[...]
order1_amount = 47.51
order_1_date = datetime(2023,9,18)
[...]
# check if order 1 is within the period
if order_1_date > start_date:
    alex_total_spend += order1_amount
[...]
```

```
>>> The total is $801.2
```

PaL

Zhou et al, PaL: Program-aided language models, ICML 2023*

Key design choices of PaL

Today is 1/20/2024, Alex made three orders: $47.51 on 9/18/2023, $765.8 on 1/1/2024, $35.4 on 1/9/2024. How much has he spent in the last 56 days?

Python

- Abundant
- Easily comprehensible

Interleave between natural language and programming language

```
order1_amount = 47.51
order2_amount = 765.8
[...]
# check if order 1 is within 56 days
[...]
```

```
a = 47.51
b = 765.8
return float(a + b)
```

[Chowdhery et al, PaLM]
[Mishra et al, Lila]
[Austin el at, Learning ..]

Zhou et al, PaL: Program-aided language models, ICML 2023*

Few-shot in-context learning with coding-proficient LLMs

Alex Martin made three orders: $47.51 on 9/18/2023, $765.8 on 1/1/2024 and $35.4 on 1/9/2024. How much he spent in my store in the last 56 days?

Input 1

Program 1

Input 2

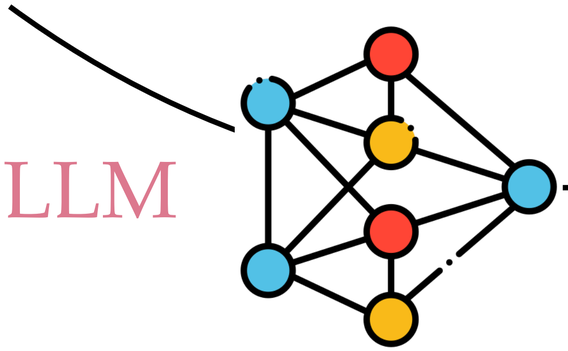
Program 2

...

In-context examples

- Manually create
- Select from a training set

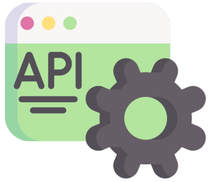
coding-proficient LLM



```
[...]
order1_amount = 47.51
order_1_date = ...
# check if [...]
```

Zhou et al, PaL: Program-aided language models, ICML 2023*

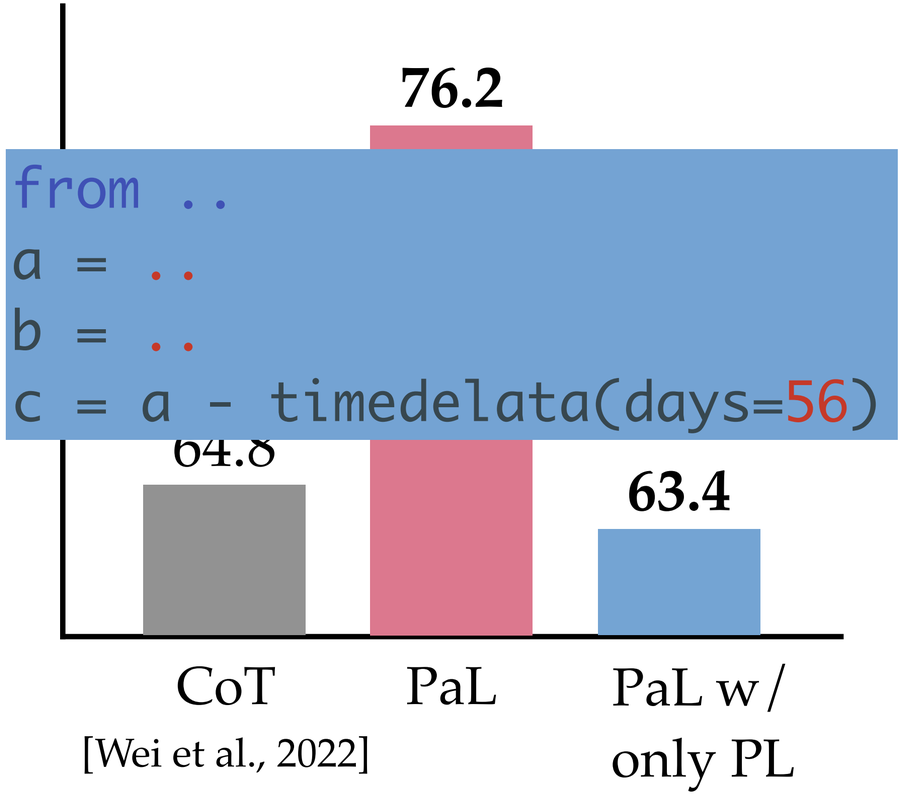
PaL offloads the solving to tools seamlessly



Today is 1/20/2024 […] How much has he spent in the last 56 days?

```
from datetime import datetime, timedelta

today = datetime(2024, 1, 20)
# calculate 56 days ago
start_date = today - timedelta(days=56)
[...]
if order_1_date > start_date:
[...]
```

Task solving accuracy (%) on date understanding (Bigbench)




```
from ..
a = ..
b = ..
c = a - timedelata(days=56)
```

[Wei et al., 2022]

[Chowdhery et al, PaLM]
[Mishra el at, Lila]
[Austin el at, Learning ..]

Zhou et al, PaL: Program-aided language models, ICML 2023*

PaL > Large language models + Tools

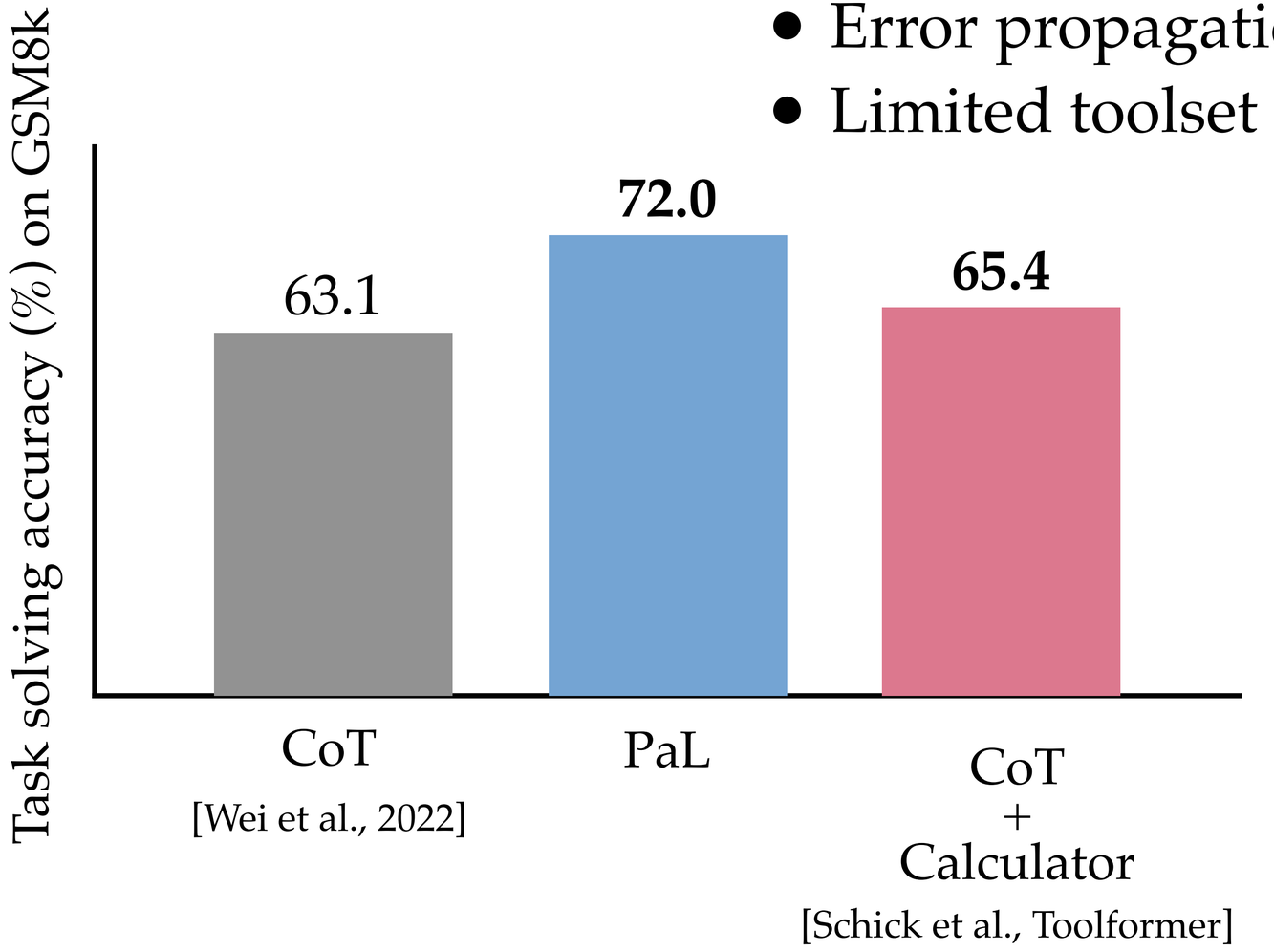
Alex made two orders within the last 56 days: one for $765.8 and another for $35.4. How much did he spend in total?

[…] the total of two orders is 765.8 + 35.8 […]

```
order1_value = 765.8
[...]
```

[…] the total of two orders is 765.8 + 35.8 **<calculator(765.8+35.8)=801.6>** 801.6[…]

- Parsing failures
- Error propagation
- Limited toolset



Zhou et al, PaL: Program-aided language models, ICML 2023*

Natural language performs example-level problem solving

Today is 1/20/2024. Alex made three orders: $47.51 on 9/18/2023, $765.8 on 1/1/2024, $35.4 on 1/9/2024. How much has he spent in the last 56 days?

Slight changes result in significant solution difference

Today is 1/20/2024. I first subtract 20 days […] The date 56 days ago is 12/20/2024
[…] Order 1 was placed on 9/18/2023, which is not within the last 56 days
[…] 765.8 + 35.4 =

Today is 2/13/2024. I first subtract 13 days […] The date 192 days ago is 8/5/2023.
[…] Order 1 was placed on 9/18/2023, which is within the last 192 days
[…] 47.51 + 765.8 + 35.4 …

Indirect

Zhou et al, PaL: Program-aided language models, ICML 2023*

Programs encourage express “task templates”

```
today = datetime(2024,1,20)
start_date = today - \
             timedelta(days=56)
[...]
if order_1_date > start_date:
    total += order_1_amount
[...]
```

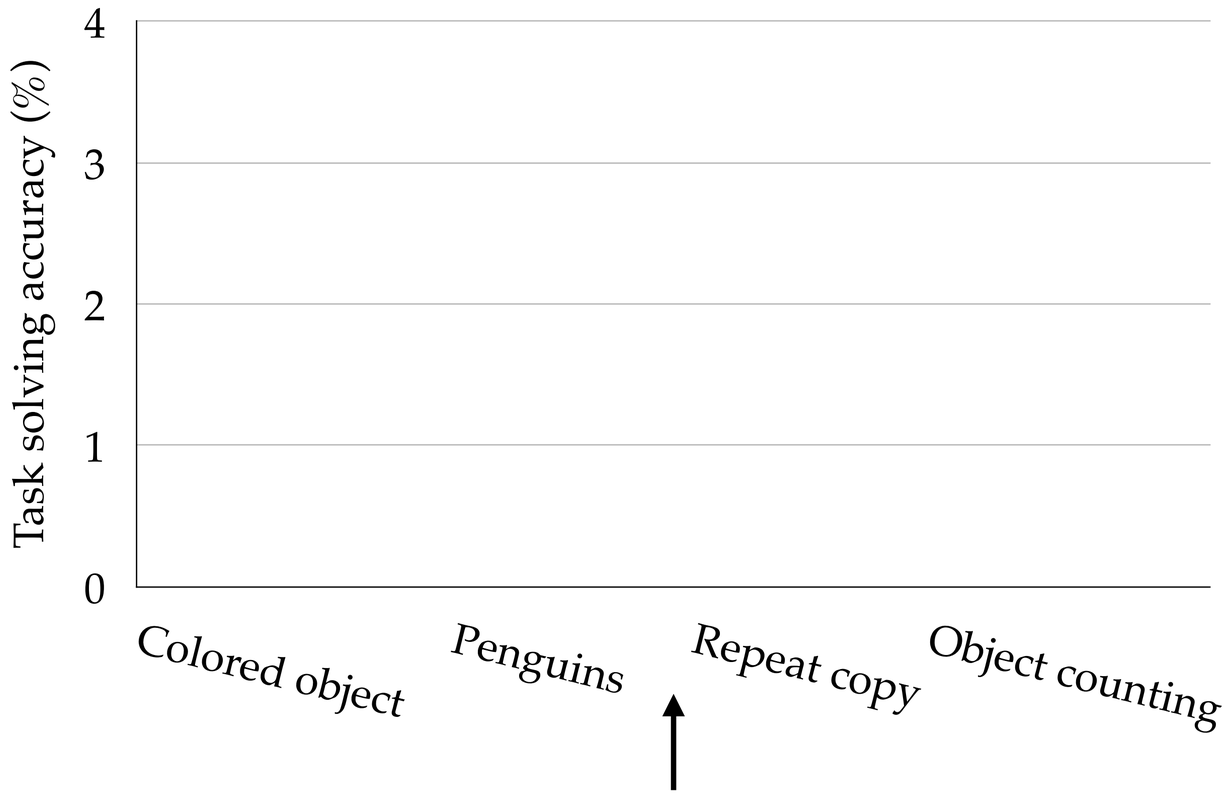
```
today = datetime(2024,2,13)
start_date = today - \
             timedelta(days=192)
[...]
if order_1_date > start_date:
    total += order_1_amount
[...]
```

direct

PaL

Zhou et al, PaL: Program-aided language models, ICML 2023*

Programs enhance LLMs in using in-context examples



Datasets where different examples share common problem-solving strategies

- Maintain an object attribute list
- Spatial reasoning

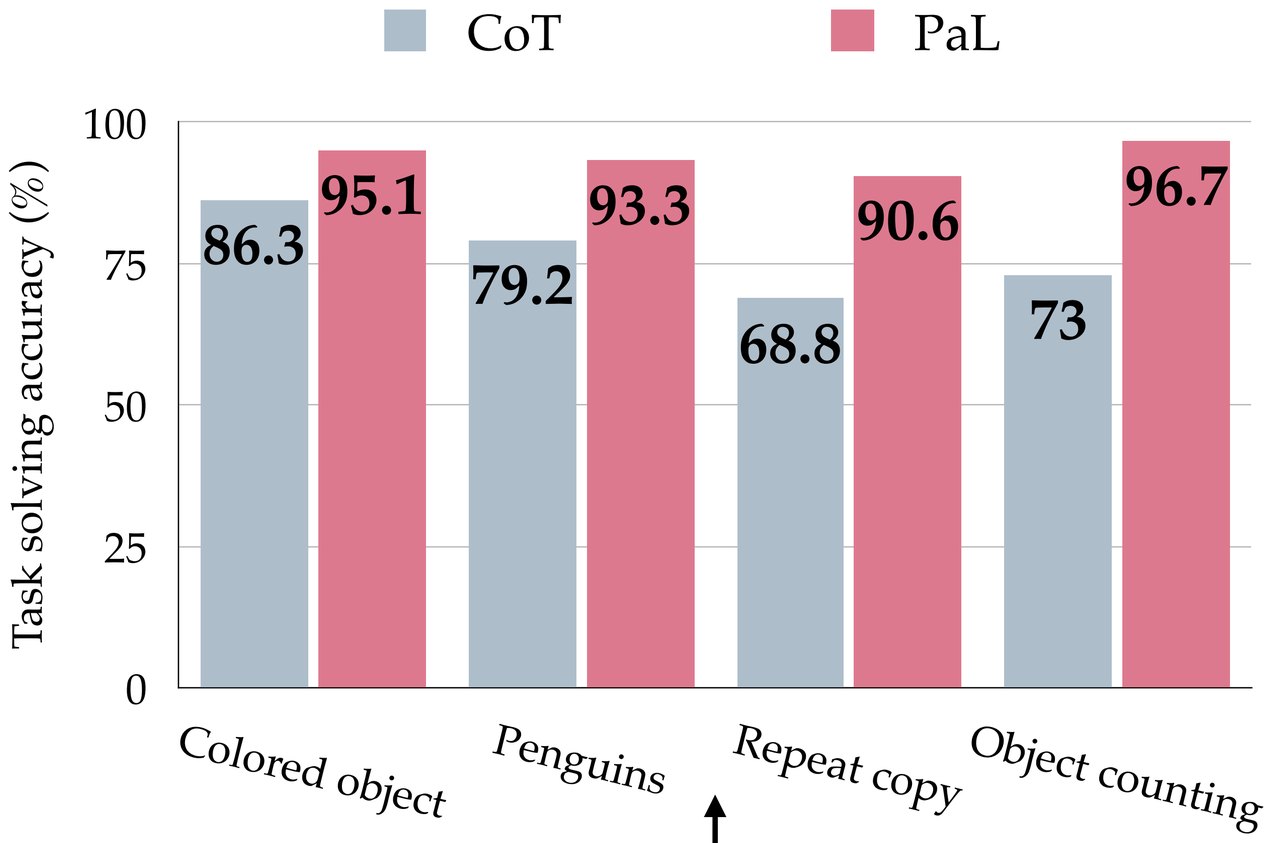
What's the color of the right most object?

What's the color of the object left to the goggle?

Example tasks in colored objects

Zhou et al, PaL: Program-aided language models, ICML 2023*

Programs enhance LLMs in using in-context examples

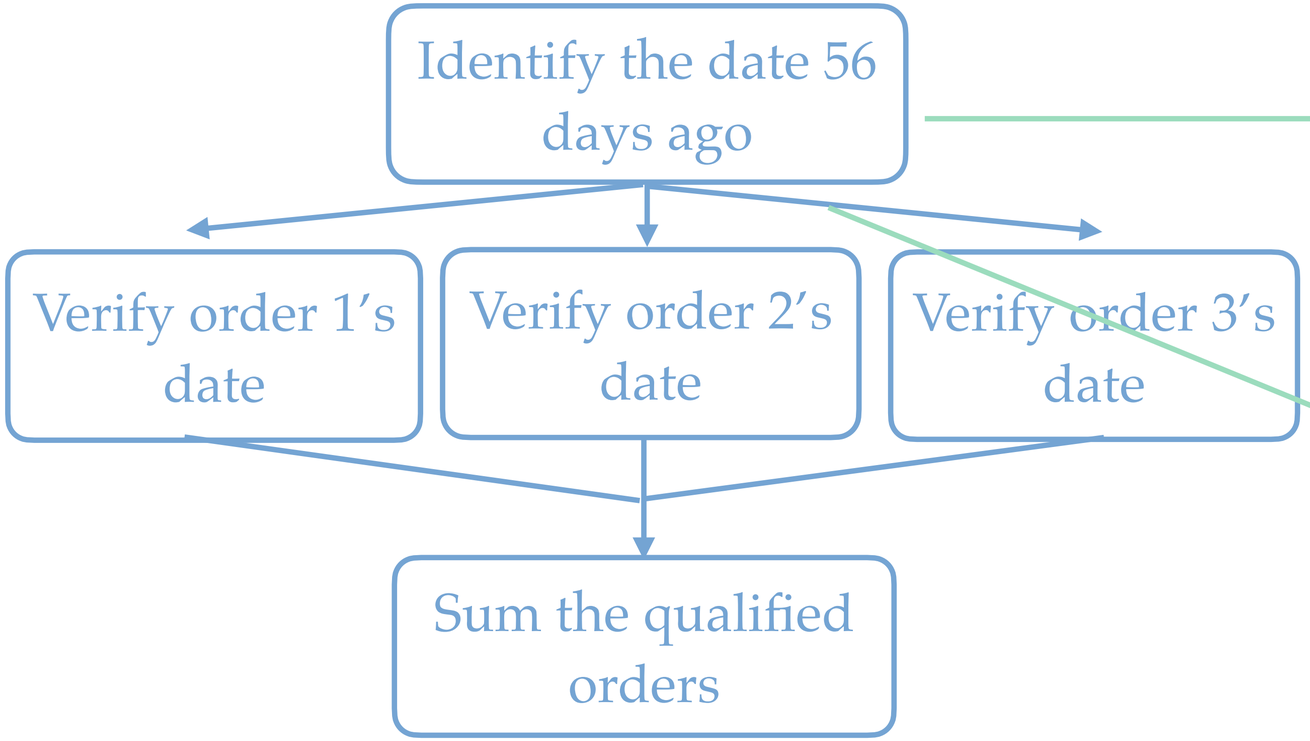


Datasets where different examples share common problem-solving strategies

Zhou et al, PaL: Program-aided language models, ICML 2023*

Bonus: Programs naturally encode structures

"Get Alex's total spend within 56 days"




```
class Graph:
  goal = "Get the total spend of
          Alex within 56 days"
  def __init__(self):
    identify_date_56_days_ago = Node()
    verify_order1_date = Node()
    [...]

    identify_date_56_days_ago.children = [
      verify_order1_date,
      verify_order2_date
      verify_order3_date
    ]
```

By a coding-proficient model

*Madaan, **Zhou** et al, Large language models of code are few-shot commonsense learners, EMNLP 2022*
*Zhang, Xu, Yang, **Zhou** et al, Causal Reasoning of Entities and Events in Procedural Texts, F-EACL 2023*

Hypothesis 1: Corpus

- Pre-training corpus for code models contains procedural knowledge useful for these tasks, e.g., game engine

```
class Flower(parentPlant:Plant) extends EnvObject {
  this.name = "flower"

def pollinate(pollen:Pollen):Boolean = {
  // Step 1A: check to see if the pollen is this plant's pollen, or a different plant's pollen
  if (pollen.parentPlant.uuid == this.parentPlant.uuid) {
    // The pollen comes from this plant -- do not pollinate
    //## println ("#### POLLEN COMES FROM SAME PLANT")
    return false
  }

  // Step 1B: Check to see that the pollen comes from the correct plant type
  if (pollen.getPlantType() != parentPlant.getPlantType()) {
    // The pollen comes from a different plant (e.g. apple vs orange) -- do not pollinate
    //## println ("#### POLLEN COMES FROM DIFFERENT TYPE OF PLANT")
    return false
  }
```

Code snippet taken from https://github.com/allenai/ScienceWorld/

Hypothesis 2: Training

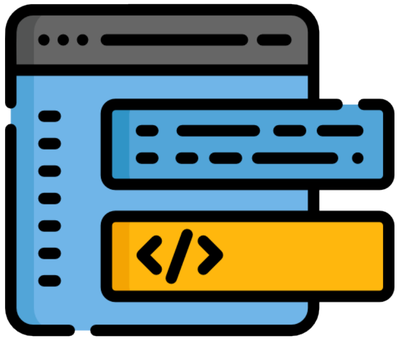
```
class BakeACake:
    def __init__(self) -> None:
        self.find_recipe = Node()
        self.gather_ingredients = Node()
        self.mix_ingredients = Node()
        self.find_recipe = Node()
        self.preheat_oven_at_375f = Node()
        self.put_cake_batter_into_oven = Node()
        self.take_cake_out_after_30_min = Node()

        self.find_recipe.children = [self.gather_ingredients, self.preheat_oven_at_375f]
        self.gather_ingredients.children = [self.mix_ingredients]
        self.mix_ingredients.children = [self.put_cake_batter_into_oven]
        self.preheat_oven_at_375f.children = [self.put_cake_batter_into_oven]
        self.put_cake_batter_into_oven.children = [self.take_cake_out_after_30_min]
```

Training on code makes the model better at
procedures / long-range inference / connecting-the-dots

[Kim et al, 2023] Coding-proficient model shows stronger performance on entity tracking

PaL brings a range of problems under one roof

Connecting PaL and follow-up work



PaL

+ Multi-sample generation
  [Zhou et al, PaL]
+ More modularized planning
  [PaL, Jiang et al]
+ Execution feedback
  [Wang et al, Sun et al]

→ Improve program generation quality

+ APIs for other modalities
  [Lu et al, Stanic et al]

→ For multi-modal tasks

+ Finetune with program-aided solution for specific domains (e.g., math)
  [Yue et al, Xu et al]

→ Sophisticated domain models

Evaluating AI agents

Speak general-purpose programming language with coherent model

Tool use

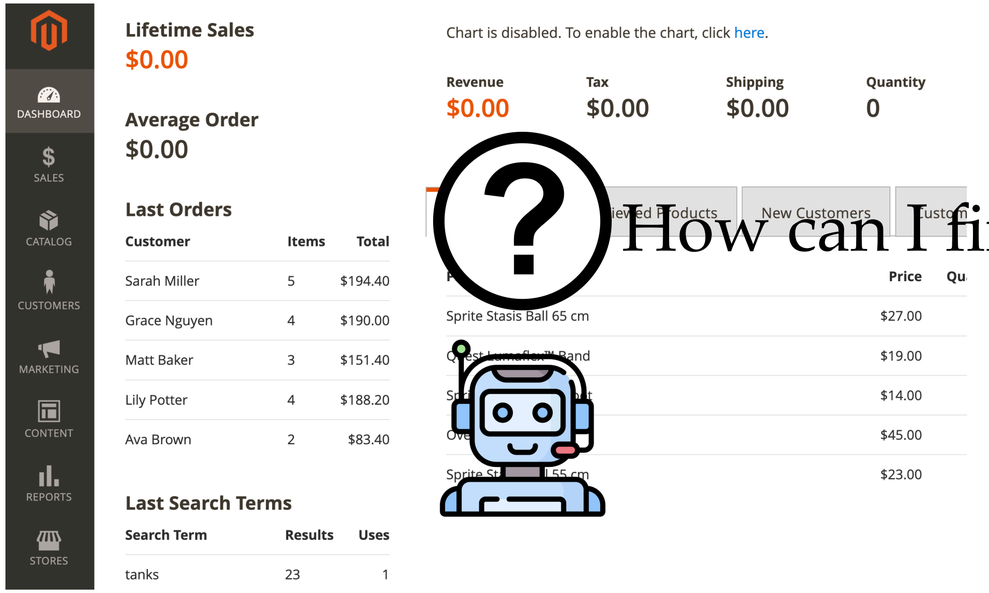
Abstract reasoning

Speaking AI's "language"

Learning by reading docs

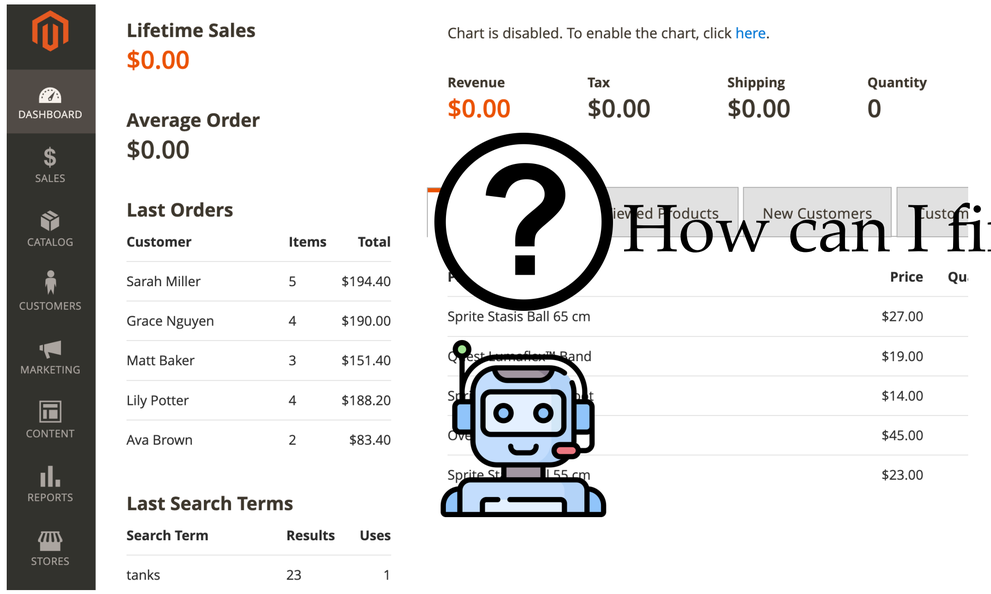
LLMs do not always have enough knowledge

Find the customer who has spent the most money in my store over the past 56 days. Send the customer some flowers.



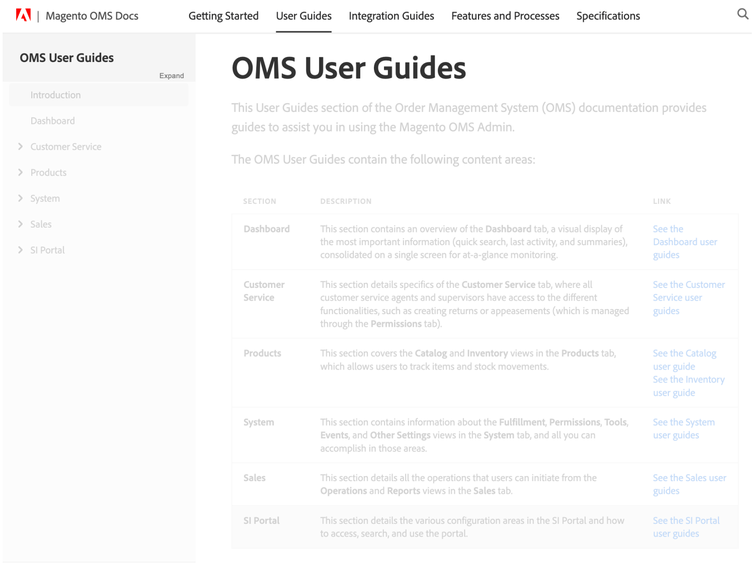
How can I find all orders?

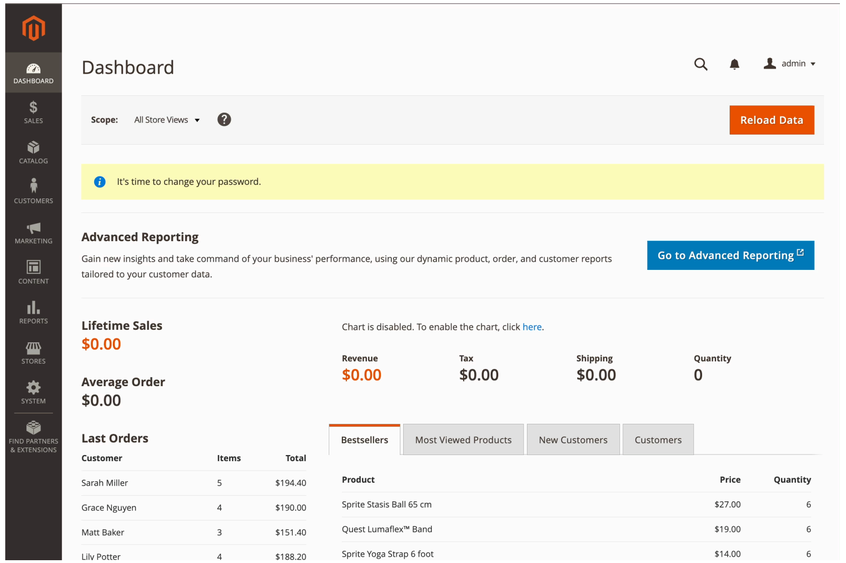
Knowledge is limited by the training cutoff

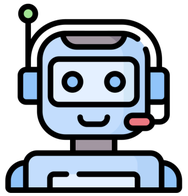


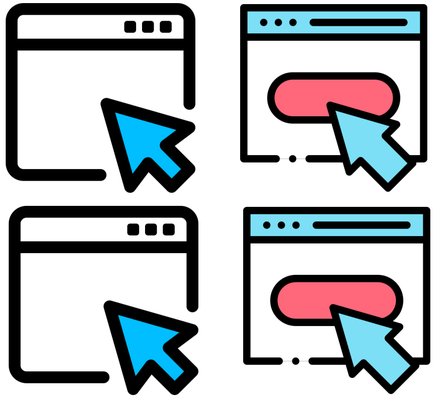
How can I find all orders?

Humans adapt to new knowledge via reading









Direct demonstrations

Not available for new knowledge

Study scenario: using new tools by reading tool docs

Bash commands

squeue
ls

Python APIs

mkdtemp
numpy

"List slurm jobs submitted by John"

"Make a temporary file to save the logs"

```
SYNOPSIS
       squeue [OPTIONS...]

DESCRIPTION
       squeue is used to view job and job step  information  for
       jobs managed by Slurm.

OPTIONS
       -A <account_list>, --account=<account_list>
              Specify  the accounts of the jobs to view. Accepts
              a comma separated list of account names. This  has
              no effect when listing job steps.

       -a, --all
              Display  information  about  jobs and job steps in
              all partitions.  This  causes  information  to  be
              displayed  about partitions that are configured as
              hidden,  partitions  that  are  unavailable  to  a
              user's  group,  and  federated  jobs that are in a
              "revoked" state.
```

```
tempfile.mkdtemp(suffix=None, prefix=None, dir=None)
```

Creates a temporary directory in the most secure manner possible. There are no race conditions in the directory's creation. The directory is readable, writable, and searchable only by the creating user ID.

The user of `mkdtemp()` is responsible for deleting the temporary directory and its contents when done with it.

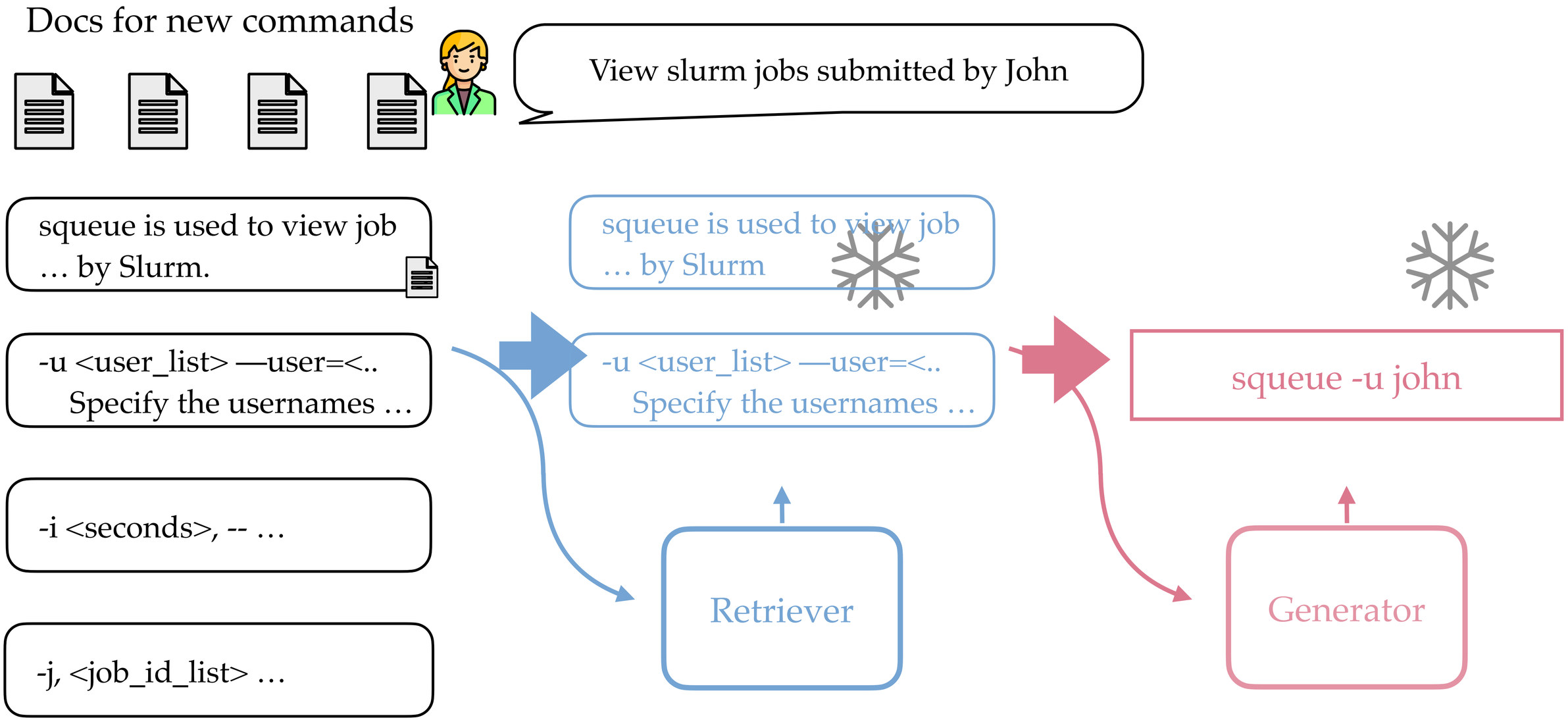
The *prefix*, *suffix*, and *dir* arguments are the same as for `mkstemp()`.

`mkdtemp()` returns the absolute pathname of the new directory.

Raises an auditing event `tempfile.mkdtemp` with argument `fullpath`.

***Zhou** et al, DocPrompting: Generating code by retrieving the docs, ICLR 2023*

DocPrompting: Retrieval-then-generation

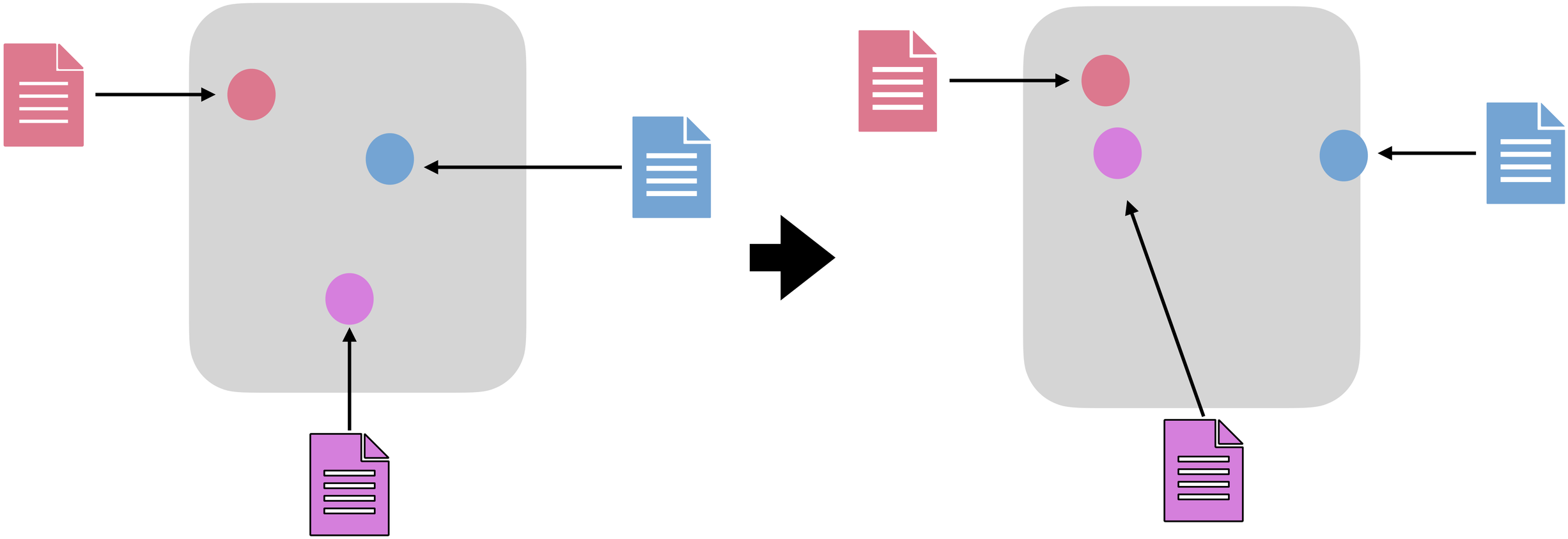


***Zhou** et al, DocPrompting: Generating code by retrieving the docs, ICLR 2023*

Contrastively training the doc retriever

$$\mathcal{L}^r = -\log \frac{\exp\Big(\mathrm{sim}(\bullet, \bullet)\Big)}{\exp\Big(\mathrm{sim}(\bullet, \bullet)\Big) + \sum_{d_j^- \in \mathcal{B}/\mathcal{D}_n^*} \exp\Big(\mathrm{sim}(\bullet, \bullet)\Big)}$$

Cosine similarity

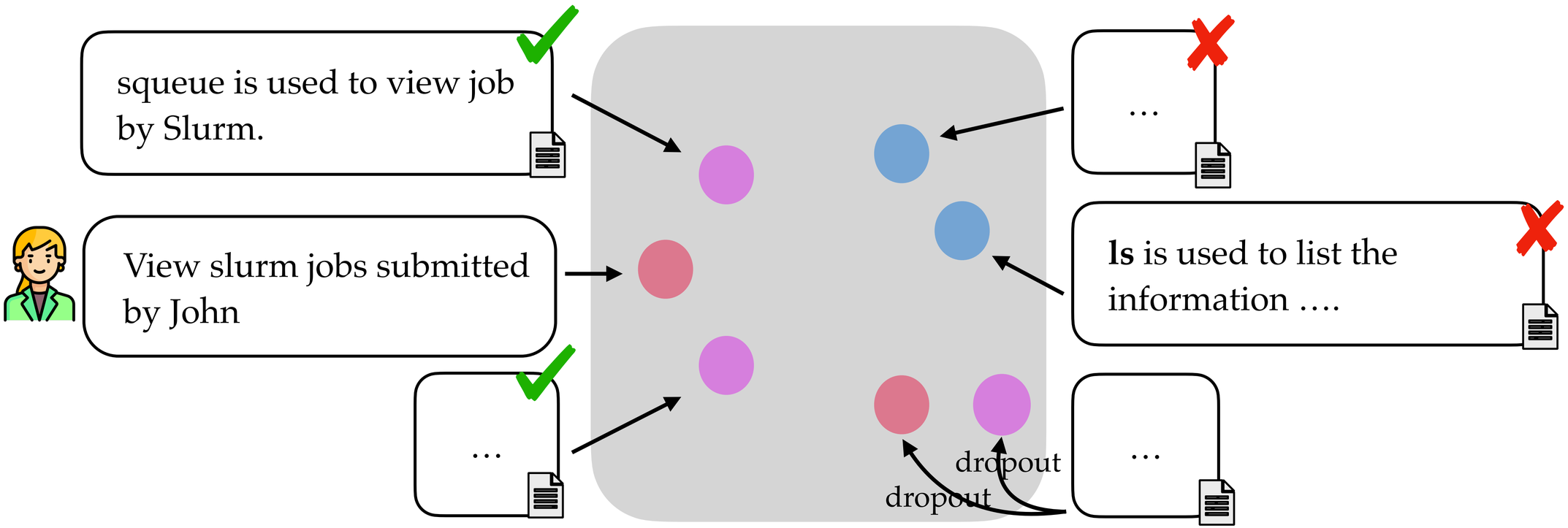


Zhou et al, DocPrompting: Generating code by retrieving the docs, ICLR 2023

Contrastively training the doc retriever

$$\mathcal{L}^r = -\log \frac{\exp\left(\text{sim}(\bullet, \bullet)\right)}{\exp\left(\text{sim}(\bullet, \bullet)\right) + \sum_{d_j^- \in \mathcal{B}/\mathcal{D}_n^*} \exp\left(\text{sim}(\bullet, \bullet)\right)}$$

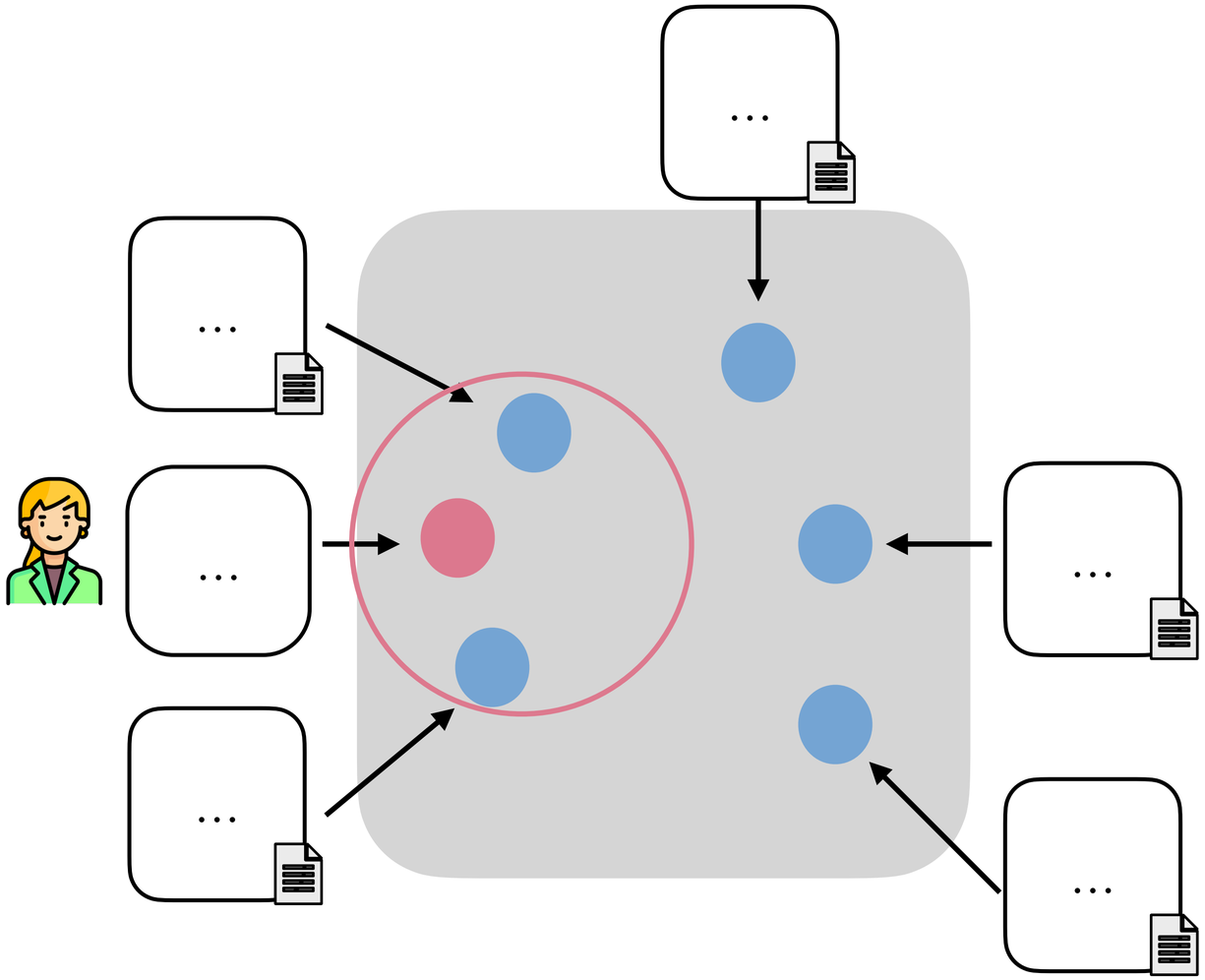
Cosine similarity



Zhou et al, DocPrompting: Generating code by retrieving the docs, ICLR 2023

[SimCSE, Gao et al.]

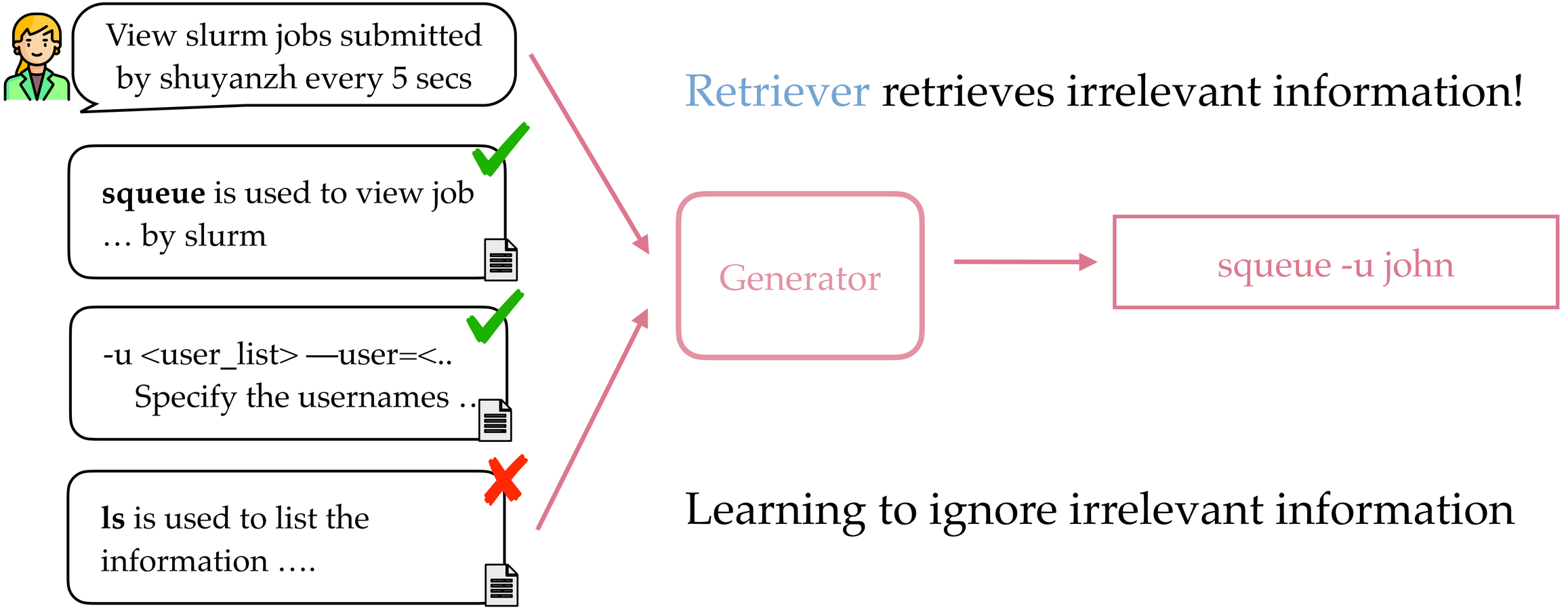
Retrieve k nearest documents



***Zhou** et al, DocPrompting: Generating code by retrieving the docs, ICLR 2023*

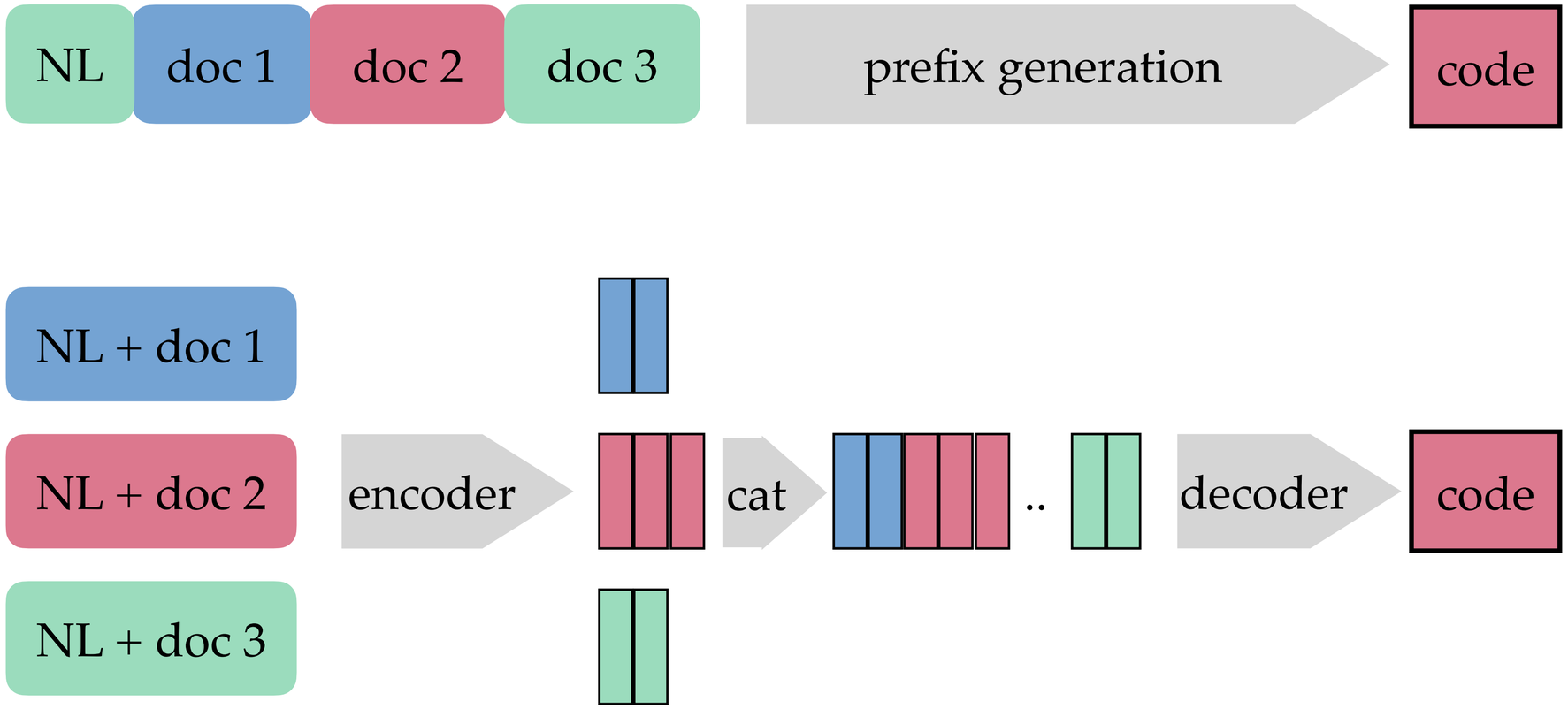
Learning to read the documents

$\log p(c * \mid$  $)$



Zhou et al, DocPrompting: Generating code by retrieving the docs, ICLR 2023

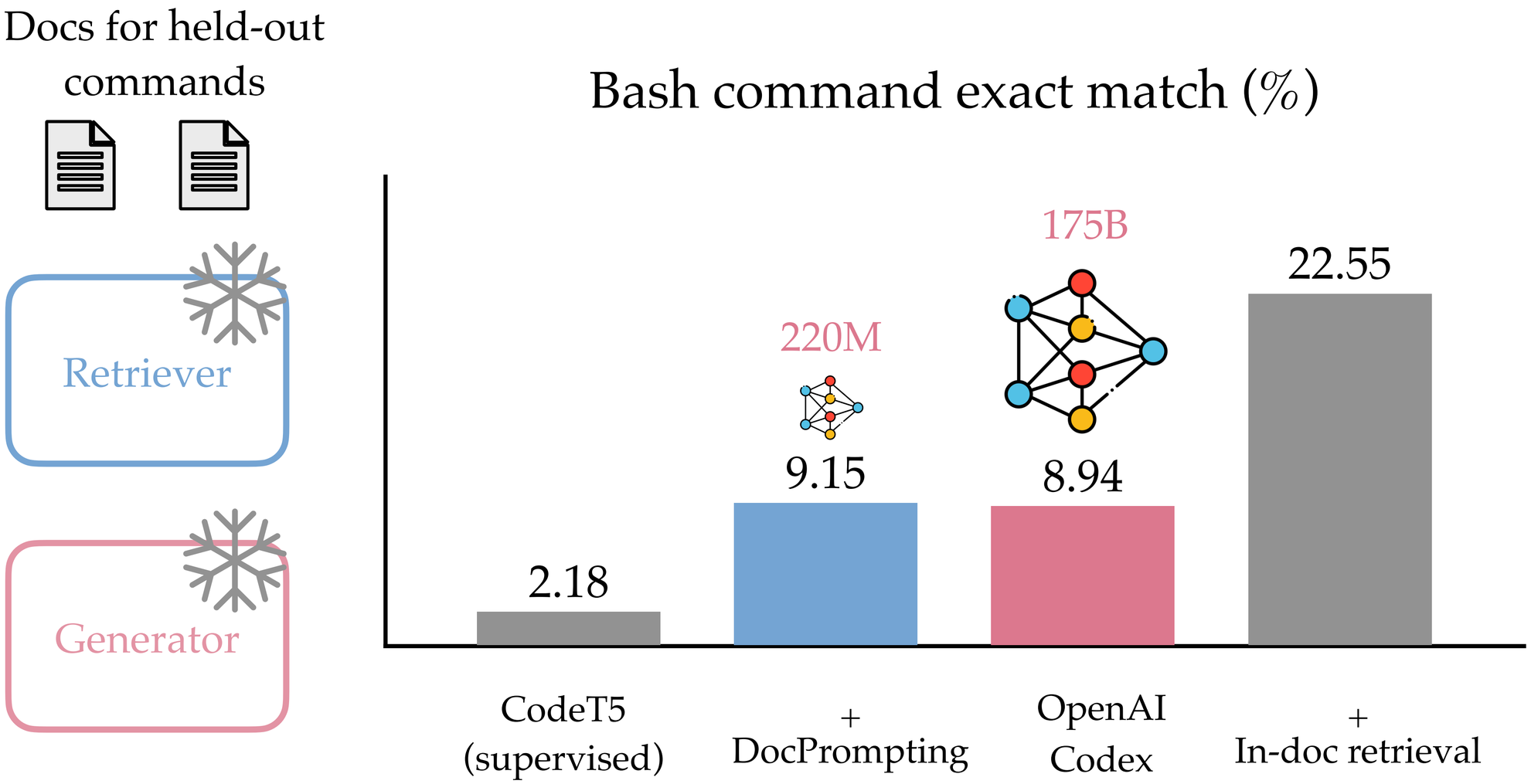
DocPrompting is applicable to various model architectures



Zhou, *Alon, Xu, Wang, Jiang, Neubig, DocPrompting, ICLR 2023*

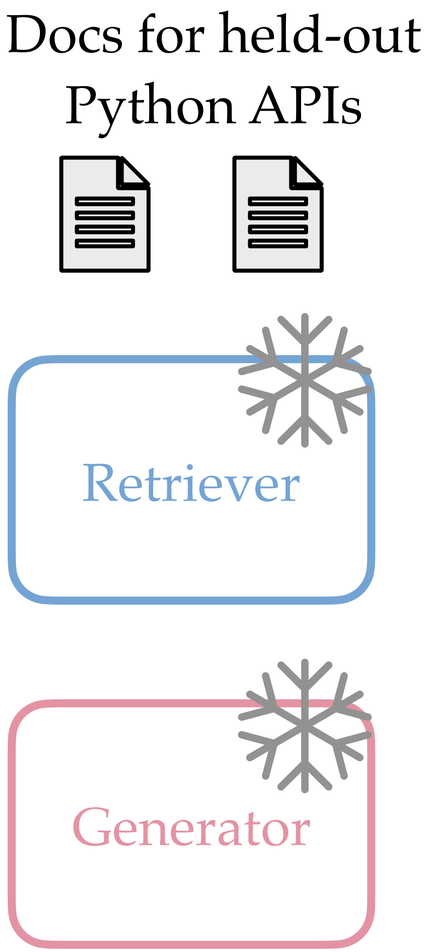
[FID, Izacard and Grave]

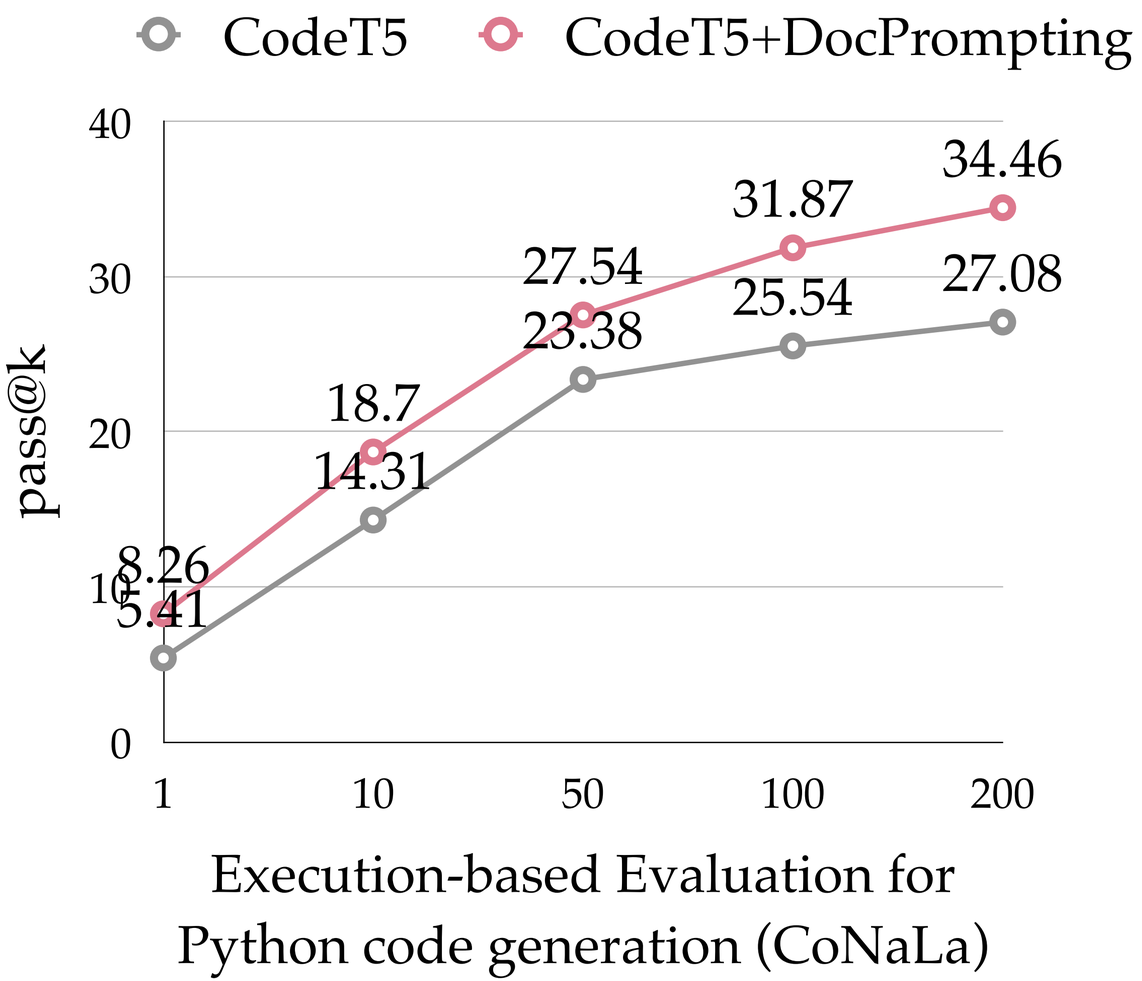
DocPrompting allows models to adapt to unseen tools without explicit demonstrations



Zhou et al, DocPrompting: Generating code by retrieving the docs, ICLR 2023

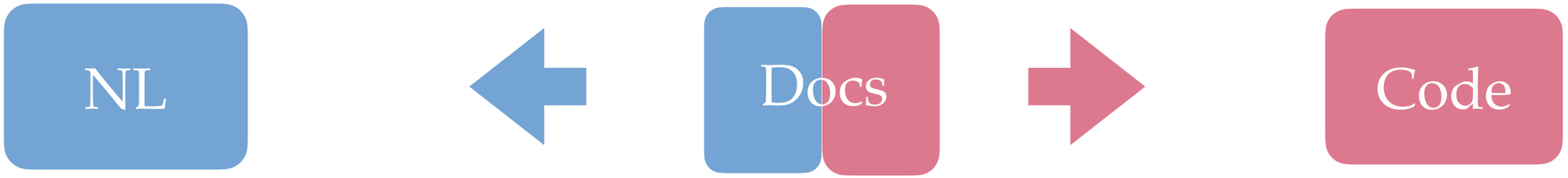
DocPrompting allows models to adapt to unseen tools without explicit demonstrations

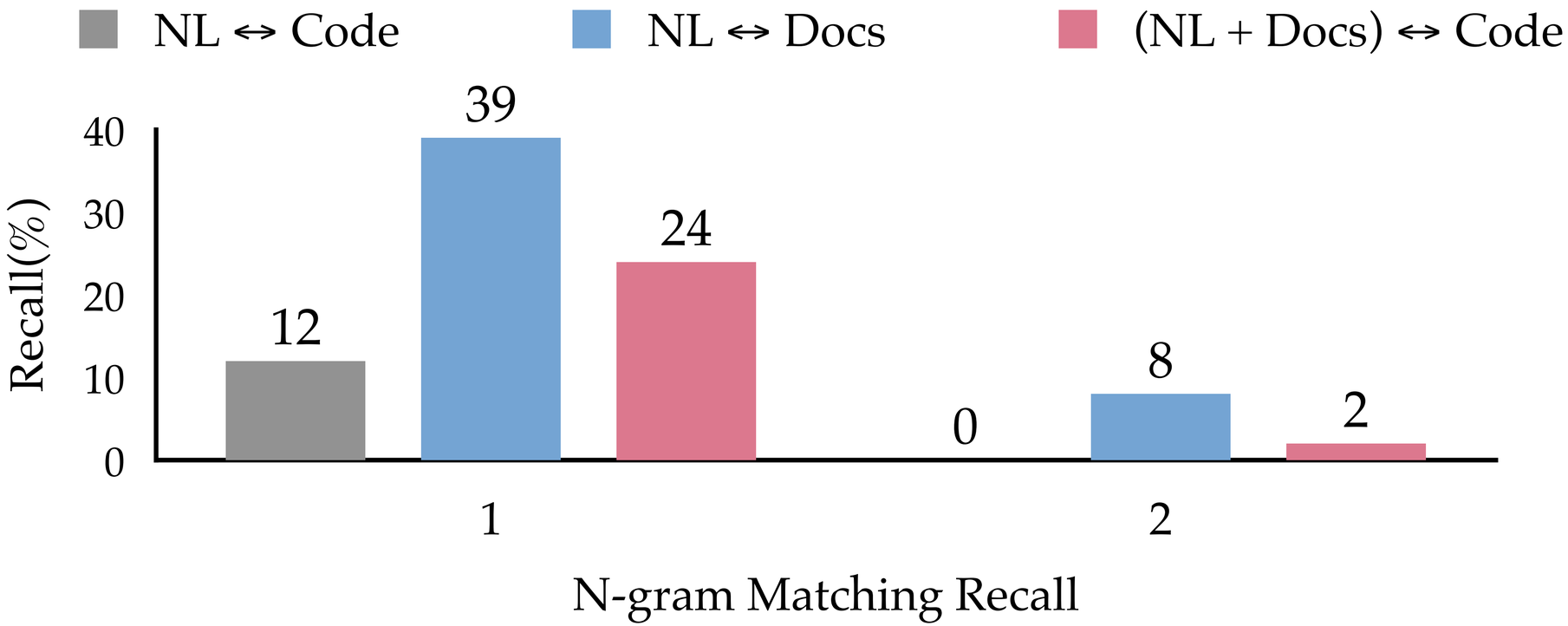




Zhou et al, DocPrompting: Generating code by retrieving the docs, ICLR 2023

Docs ease the mapping between NL and code



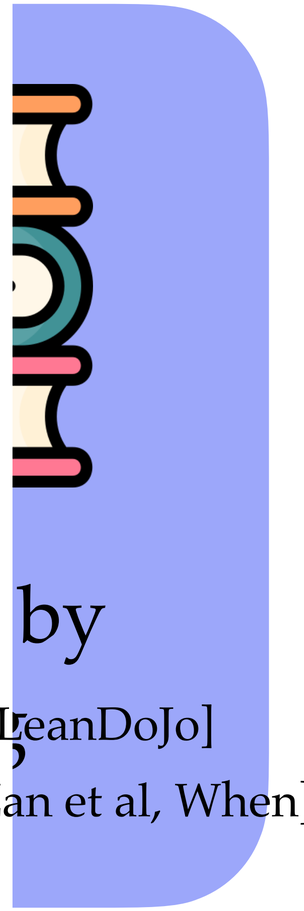


***Zhou** et al, DocPrompting: Generating code by retrieving the docs, ICLR 2023*

Evaluating AI agents

What docs created by humans that explain the tool usage

How retrieval and doc-augmented generation



✅ Up-to-date knowledge

Human-written docs as learning resources

+ Code document generation

- **Theorem proving** [Wu et al, LeanDoJo]
- **Proprietary code libraries** [Zan et al, When]
- **API use in products**

- [**Zhou** et al, Generating Code Explanations with Controllability on Purpose]