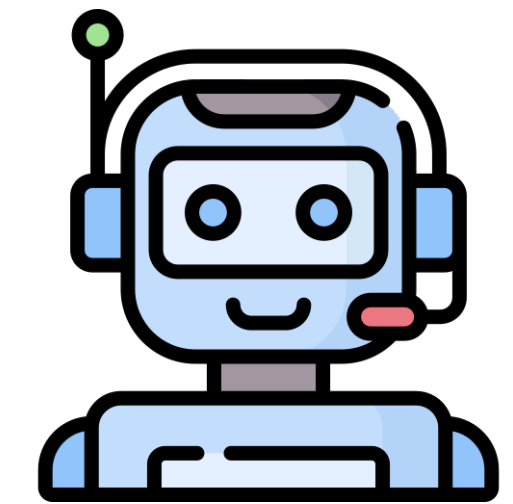




Solving Real-World Tasks with AI Agents



Shuyan Zhou

Language Technologies Institute

Carnegie Mellon University

shuyanzh@cs.cmu.edu

shuyanzhou.com



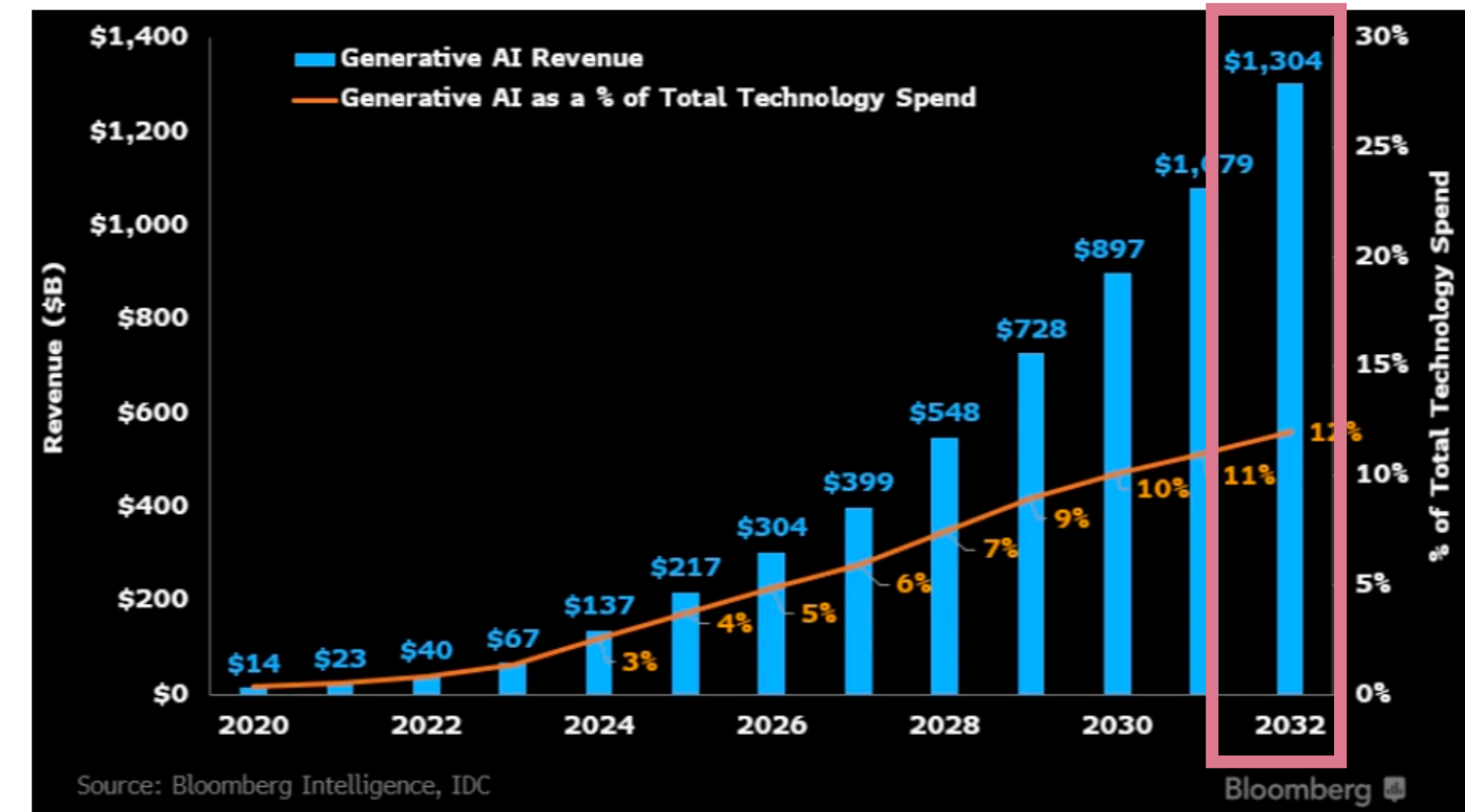
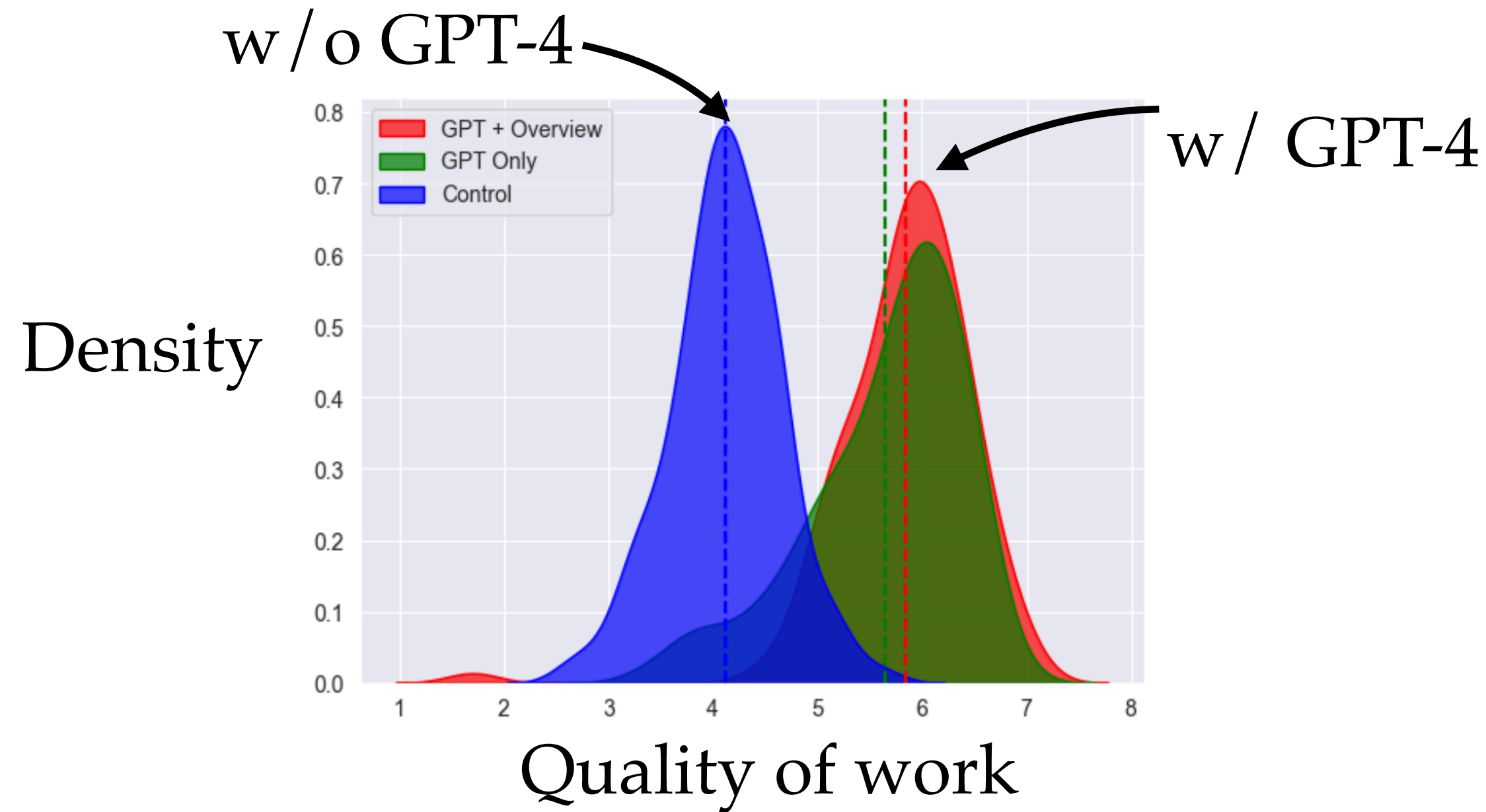
Carnegie Mellon University

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

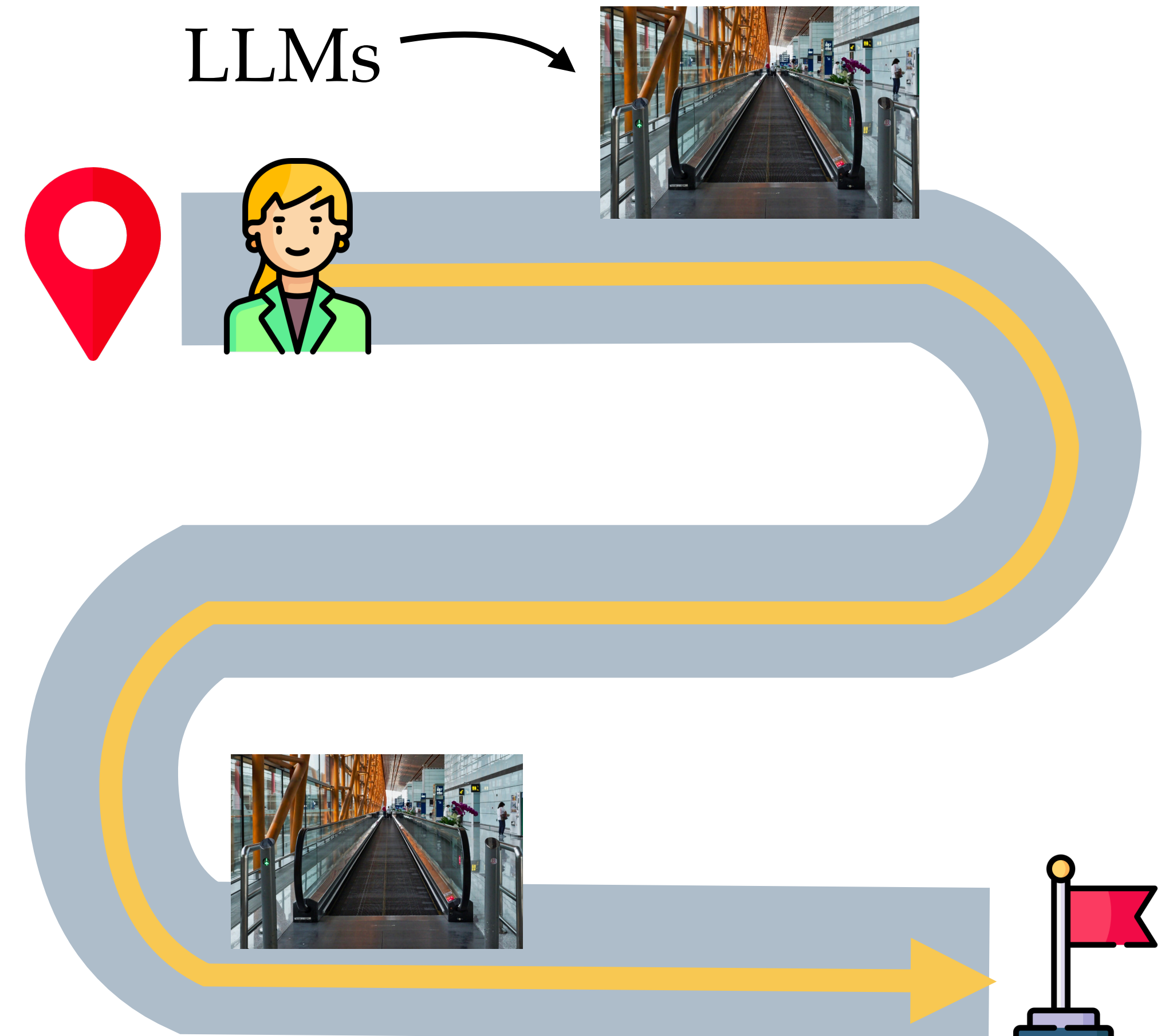
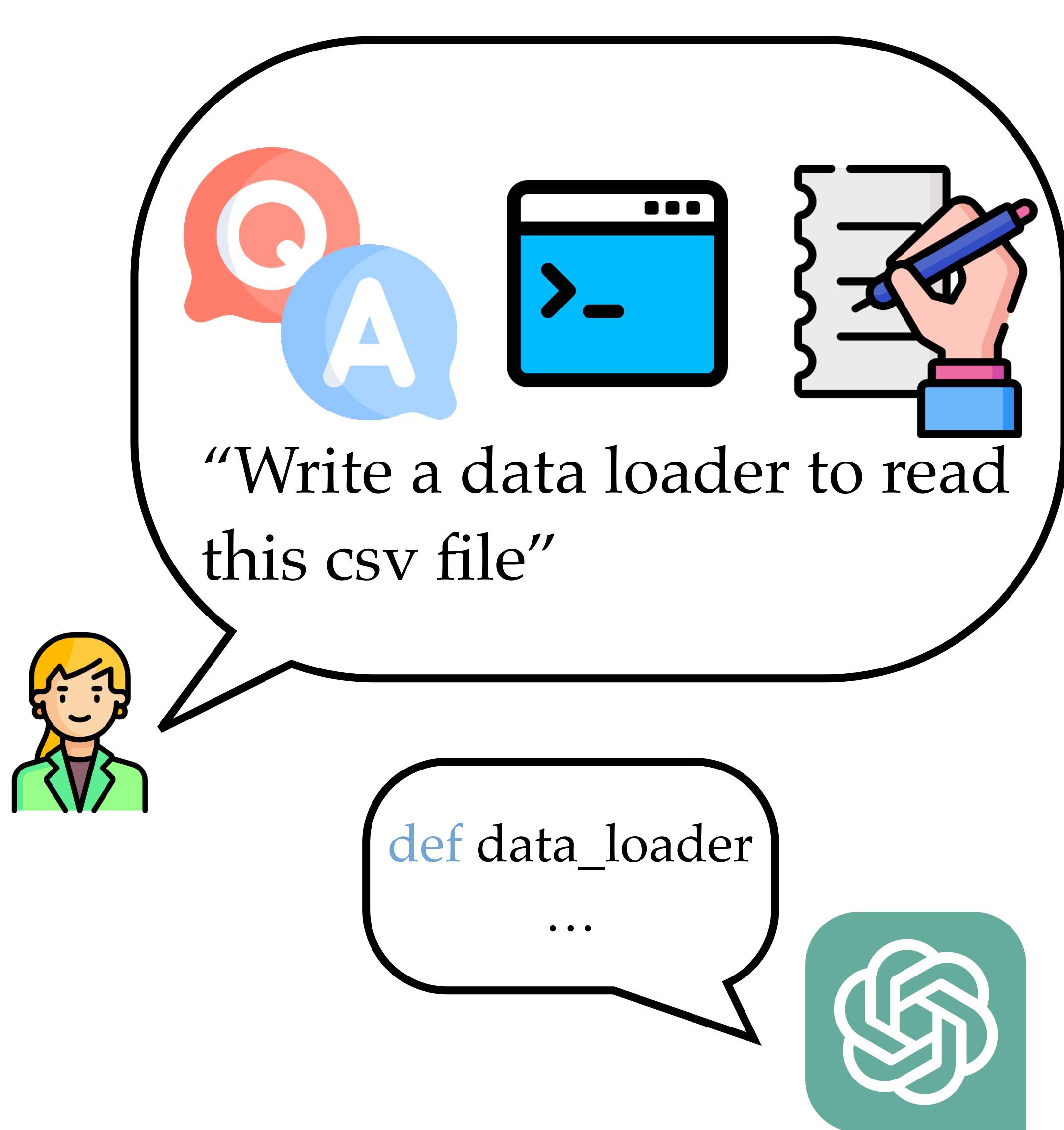


DAVOS WEF

Tech execs say a type of AI that can outdo humans is coming, but have no idea what it looks like

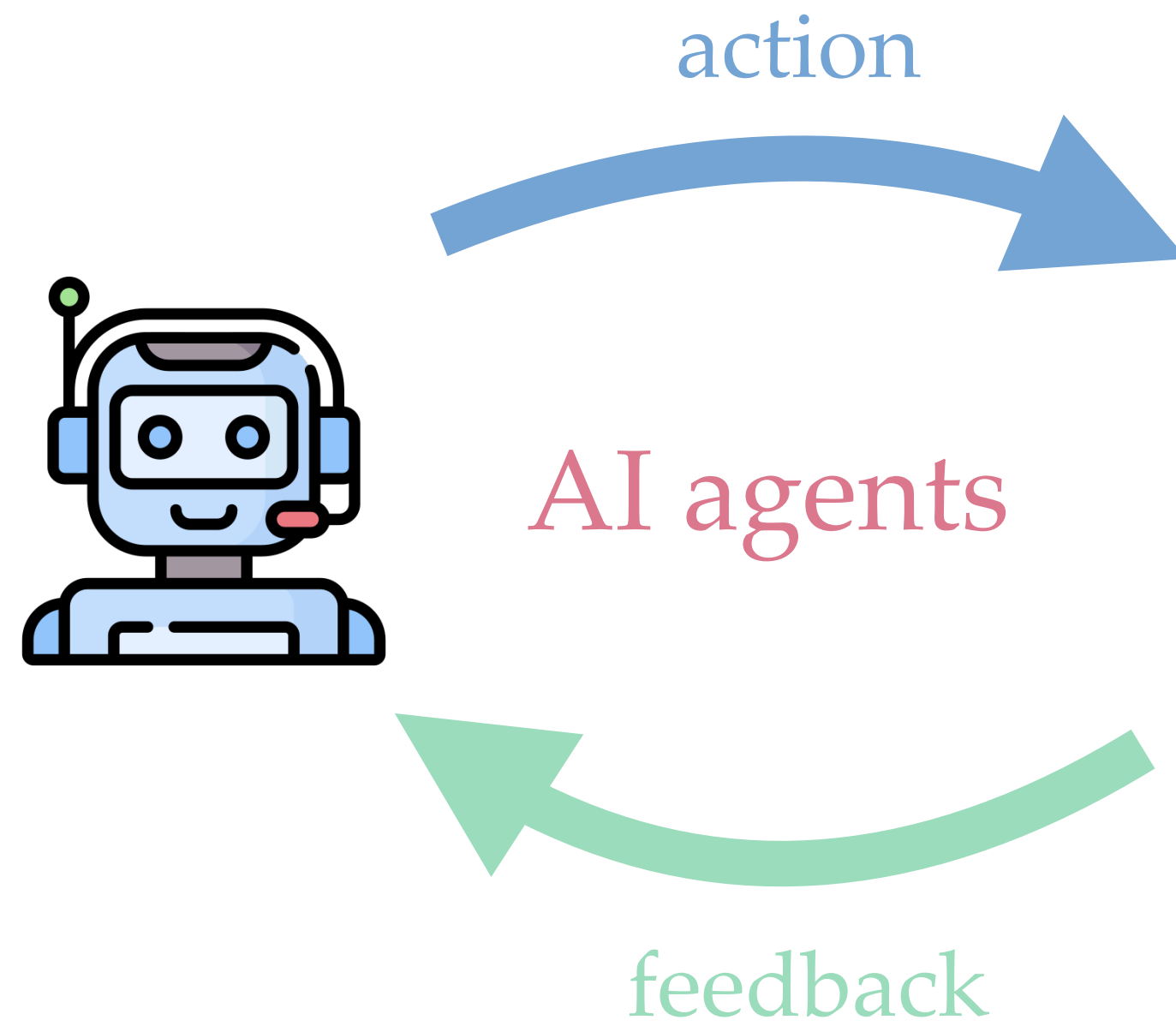
PUBLISHED TUE, JAN 23 2024 4:48 AM EST | UPDATED TUE, JAN 23 2024 9:25 AM EST

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

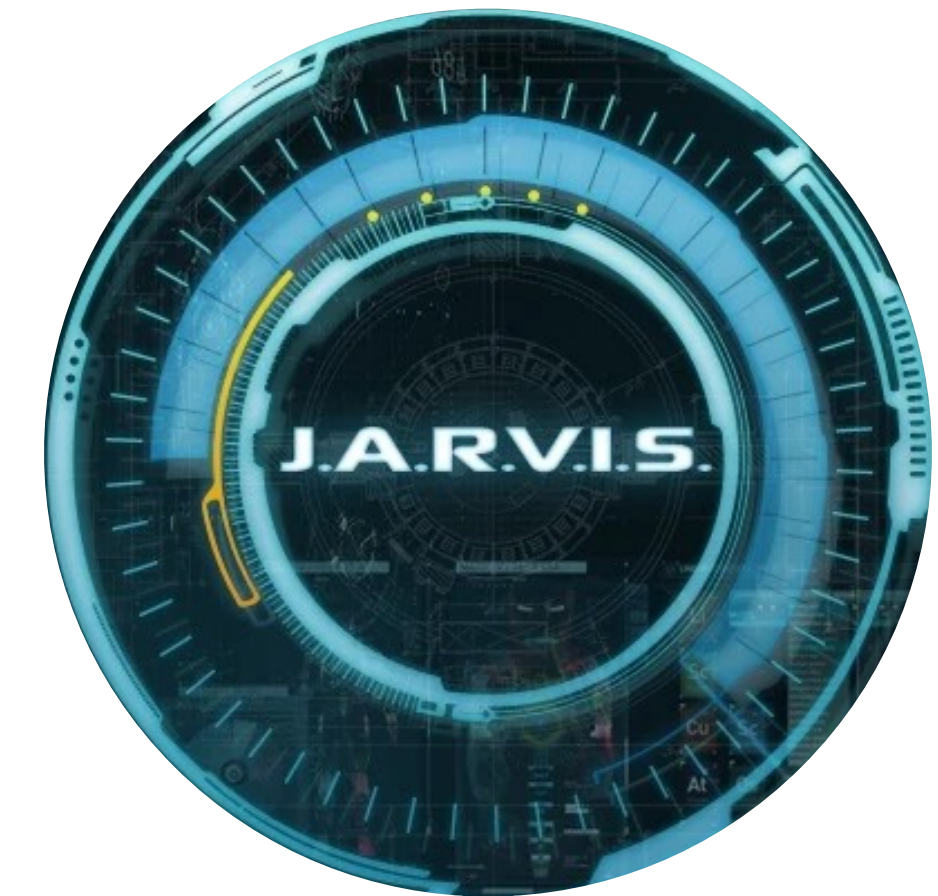
The dream of AI is far more wild



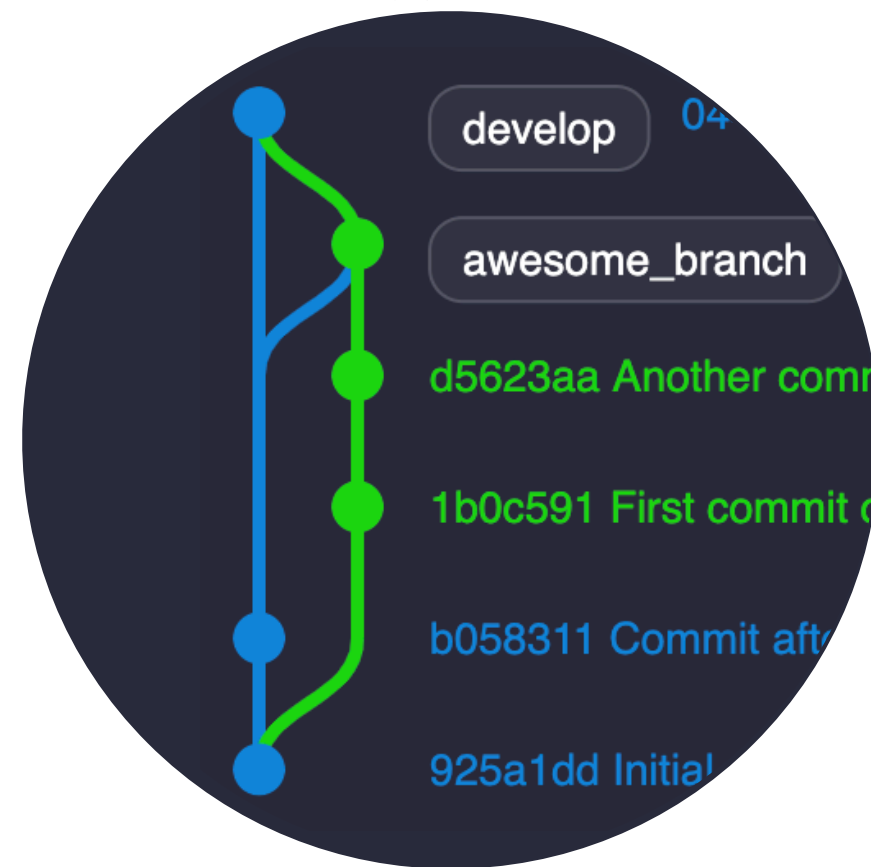
My research goal

Perform scientific research

Automate *various* tasks with minimal human intervention



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?



Evaluating AI agents

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

Natural language has inherent limitations



Speaking AI's "language"

- *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*

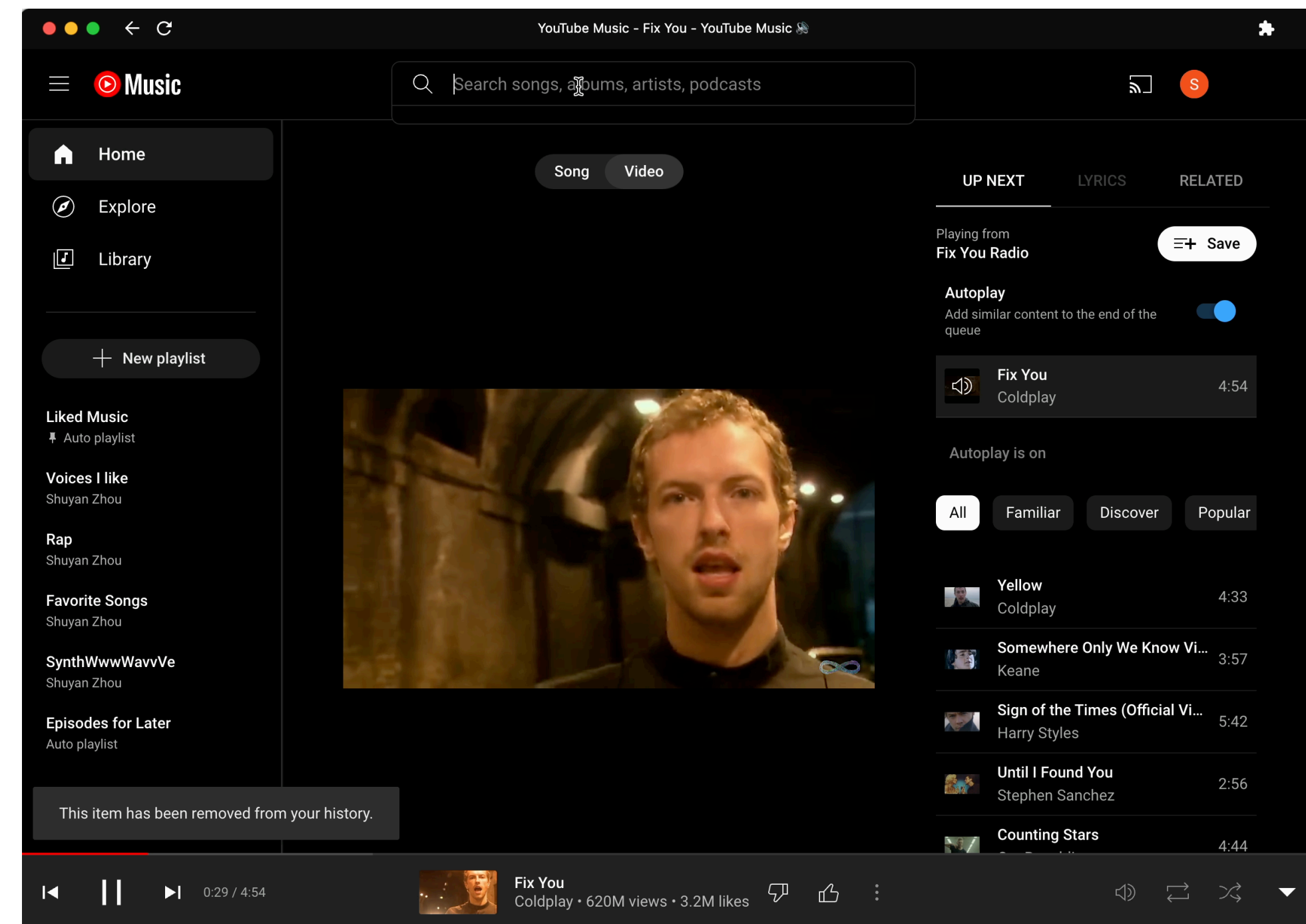
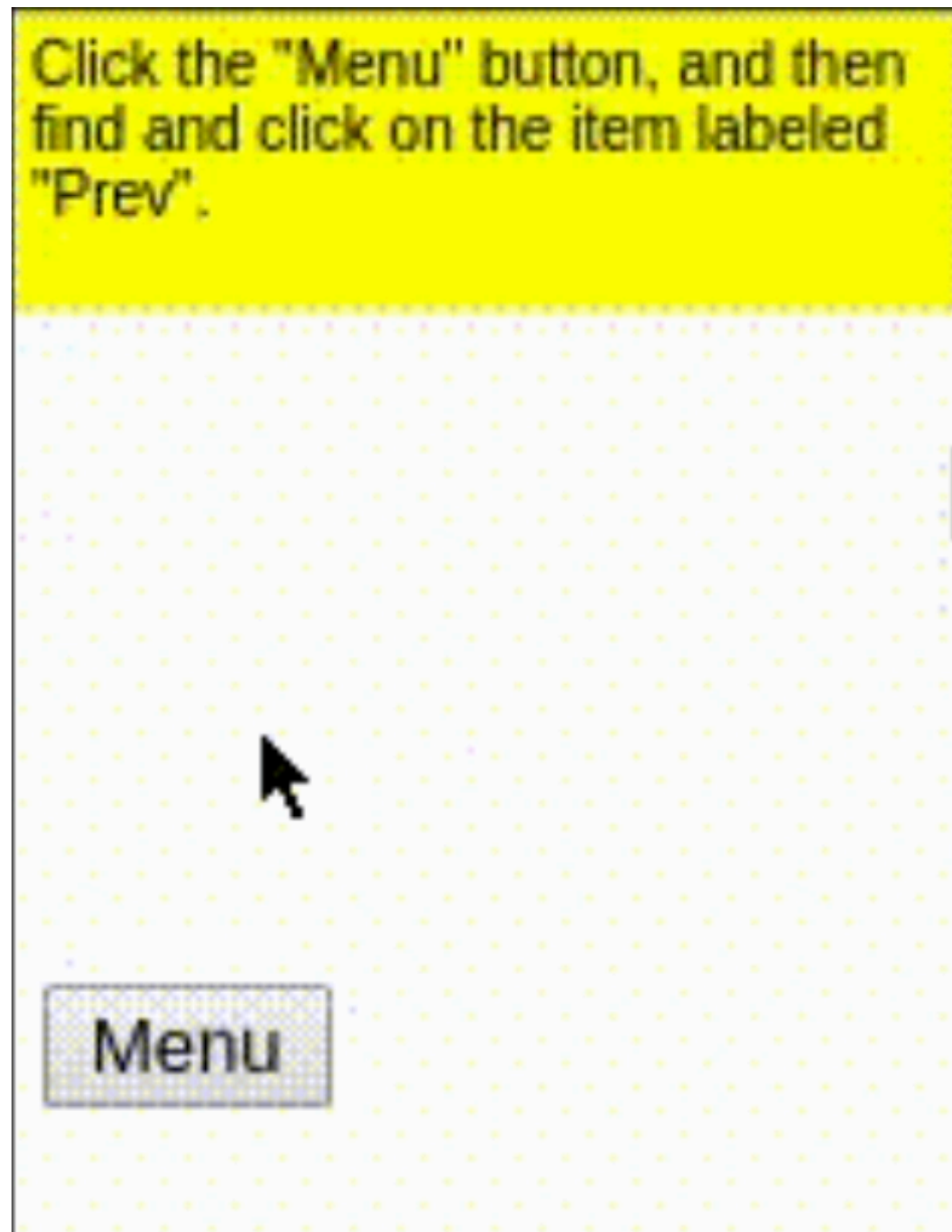
LLMs know up to a cutoff date



Learning new knowledge by reading

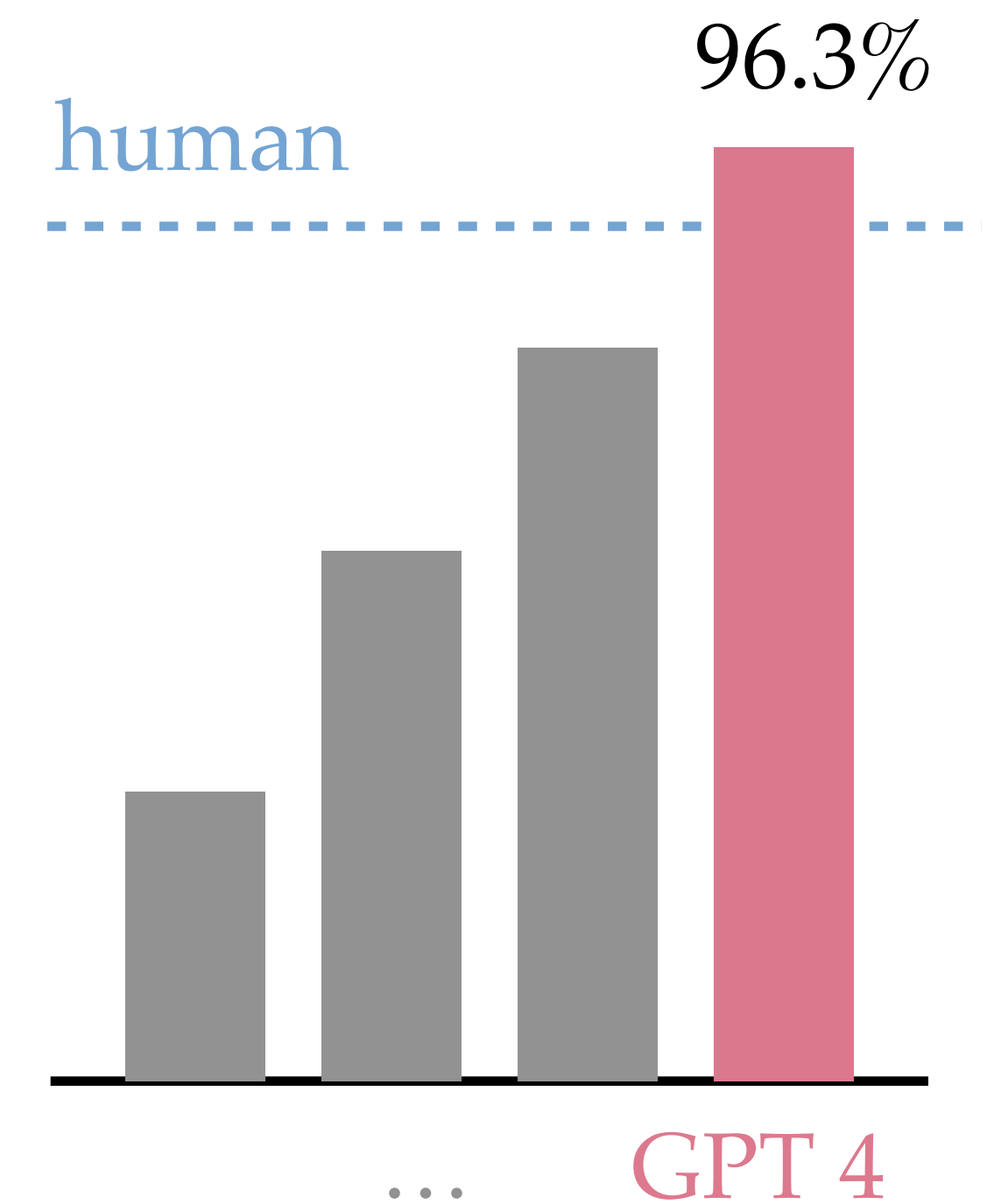
- *Zhou et al., DocPrompting, ICLR 2023*
- *Zhou* et al., Hierarchical Procedural KB, ACL 2022*

Significant gap in benchmarks vs real-world applications



"Play my favorite music"

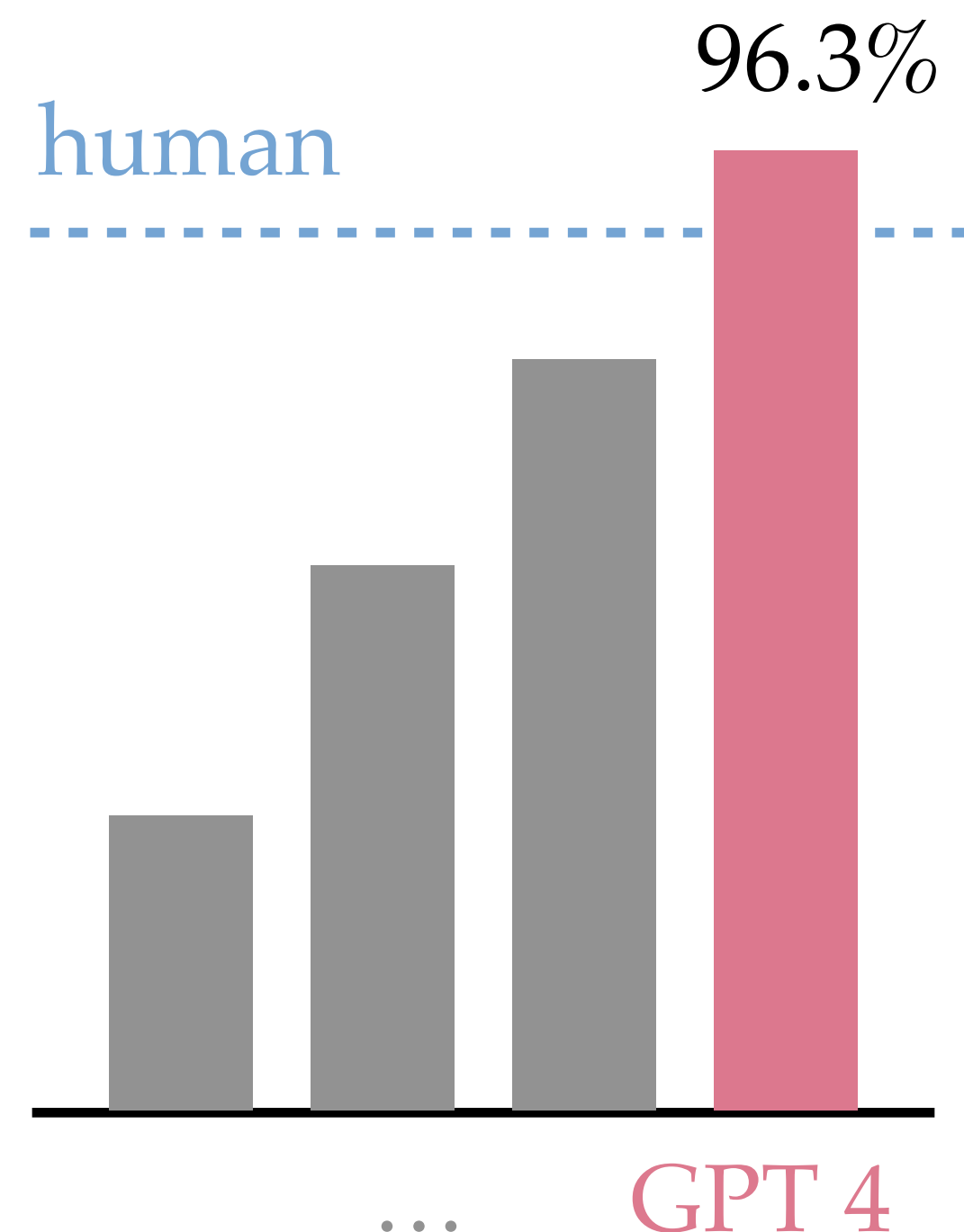
Task-solving rate on
Miniwob++



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

Significant gap in benchmarks vs real-world applications

Task-solving rate on
Miniwob++



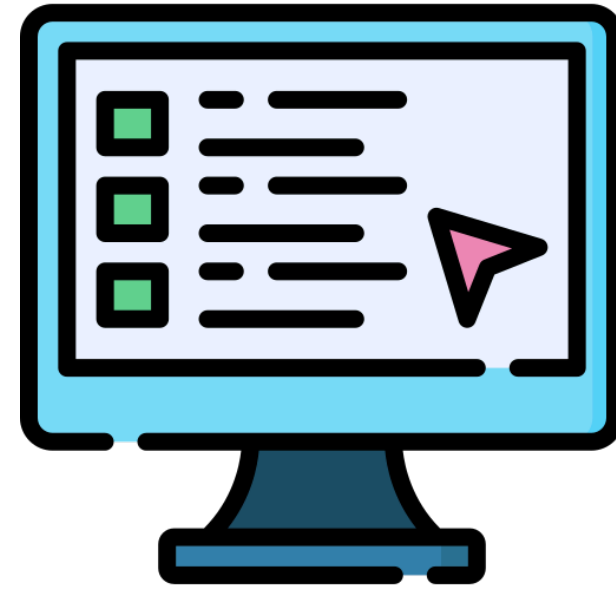
The screenshot shows a GitLab issue page for a bug report. The issue title is "[Bug] 404s, bad host, timeouts, bad urls for URLs linked from website". The description states: "I checked links in the website with brokenlinkcheck.com and found the following links could potentially have problems". A table lists five URLs with their corresponding line numbers. The right sidebar shows the issue's metadata, including the assignee (myself), due date (None), time tracking (No estimate or time spent), confidentiality (Not confidential), lock issue (Unlocked), and notifications (checked).

#	URL	lin
1	https://jenniferbrownconsulting.com/inclusion-the-book/	Inc Th &
2	https://www.getstark.co/newsletter	St
3	https://www.a11yproject.com/posts/everyday-accessibility/A11yProject.com/Resources	Th Re
4	https://chrome.google.com/webstore/detail/i-want-to-see-like-the-co/jebeedfnielkcjlcokhiobdkjjpbjia	La
5	https://chrome.google.com/webstore/detail/nocoffee/jjeeggmbnhckmgdhmgdckeigabjfbddl	Nc

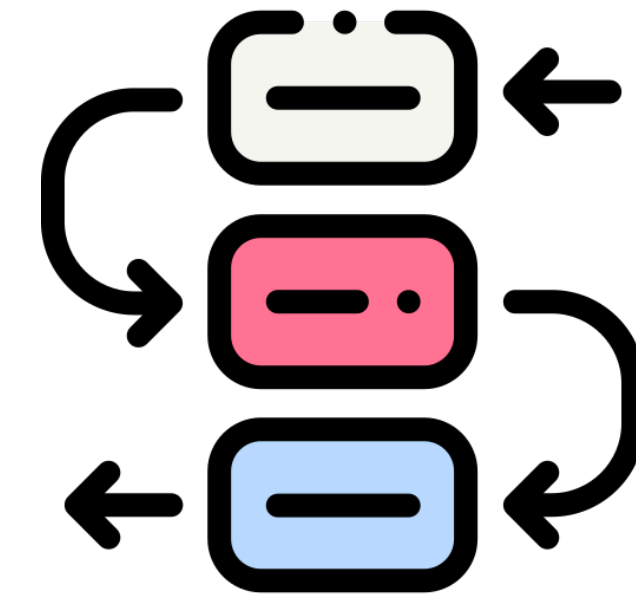
"Assign this issue to myself"

Requirements for the agent evaluation

**Realistic
interactive
environment**



**Useful &
complex
tasks**



Existing evaluations make trade-offs between them

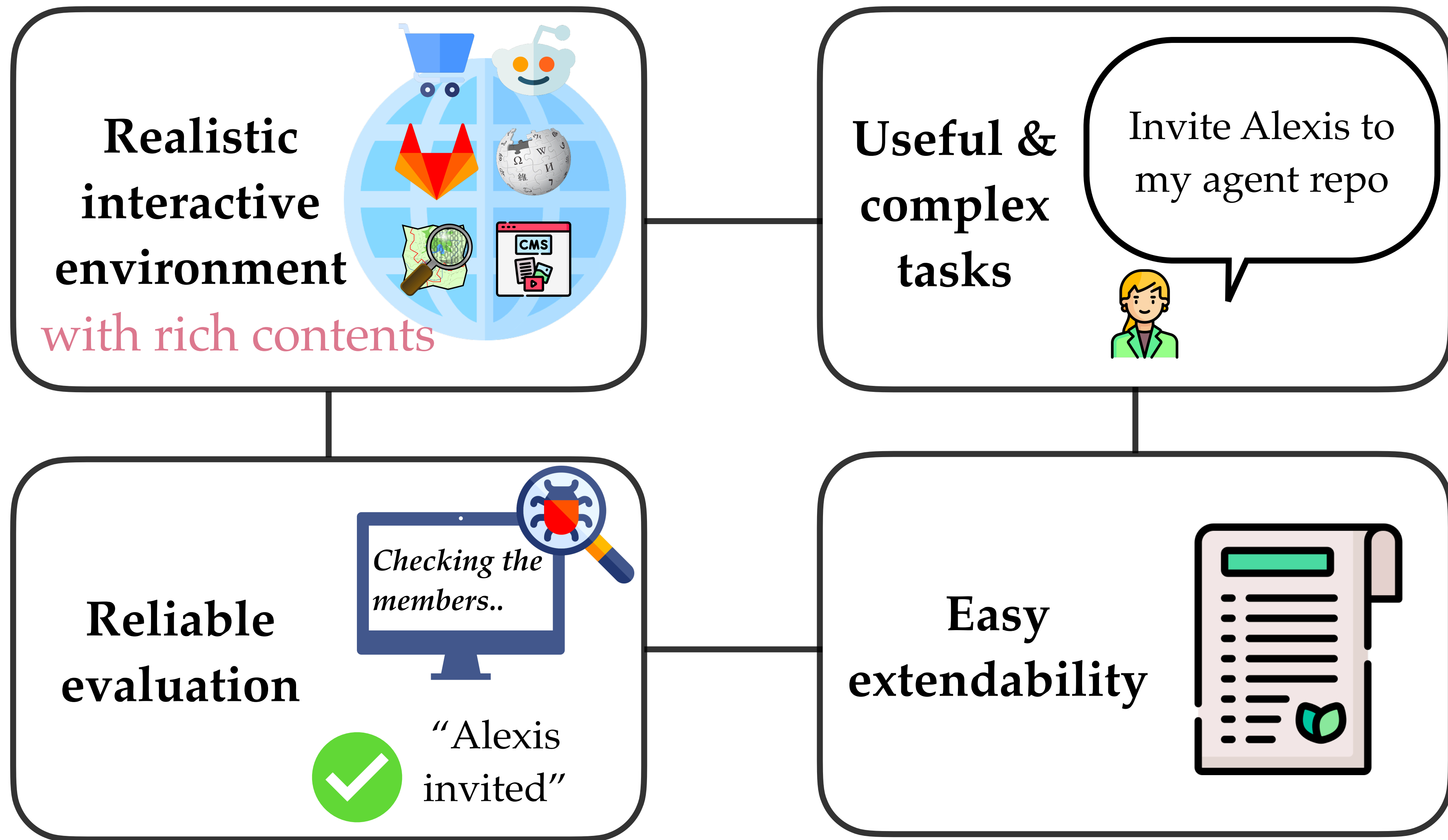
**Reliable
evaluation**



**Easy
extendability**



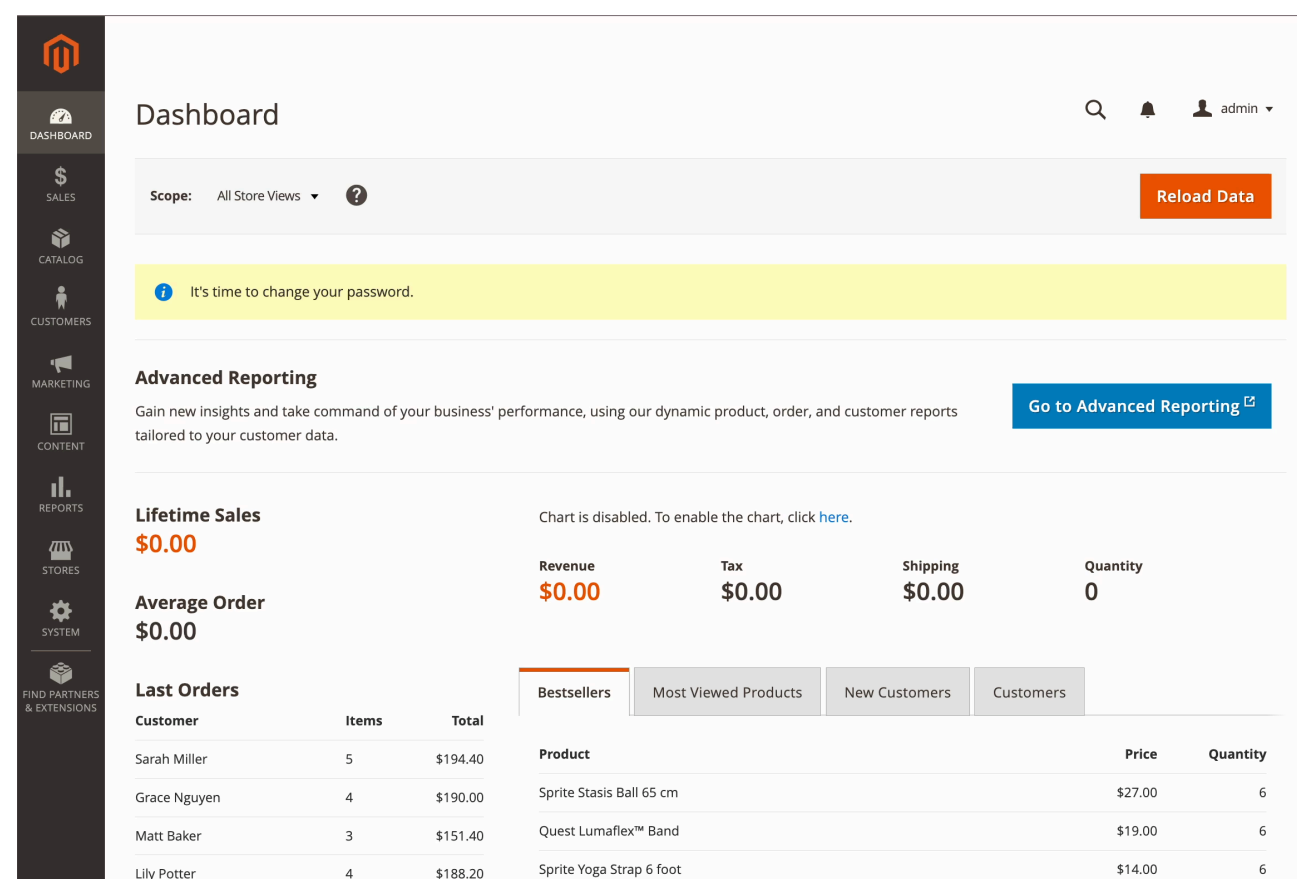
WebArena fulfills all requirements without compromise



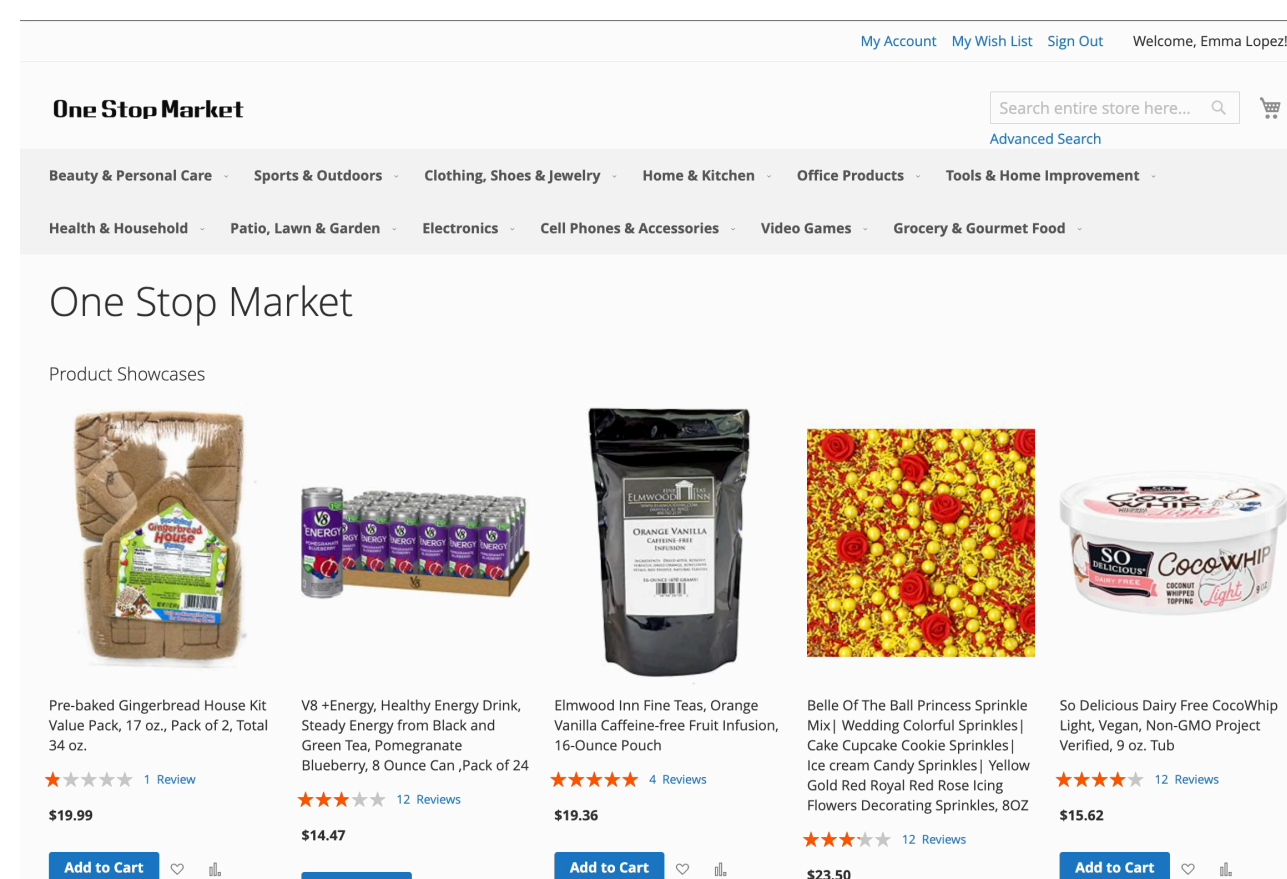
Example task in WebArena

Shop owner 

Find the customer who has placed orders worth more than \$1000 over the past 56 days.
Send the customer some **Customer appreciation task**



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



Buy some flowers online to the customer

812 long-horizon, realistic computer tasks

Outcome-based evaluation

- A new order with flowers

Order # 000000190

Product Name

ShineBear Eternal Flowers Dried Flower
Fresh Flower Live Rose Enchanted Glass
Box - (Colorful Flower Glass)

flowers

Color

Blue / Flower Glass

- Shipped to Alex Martin

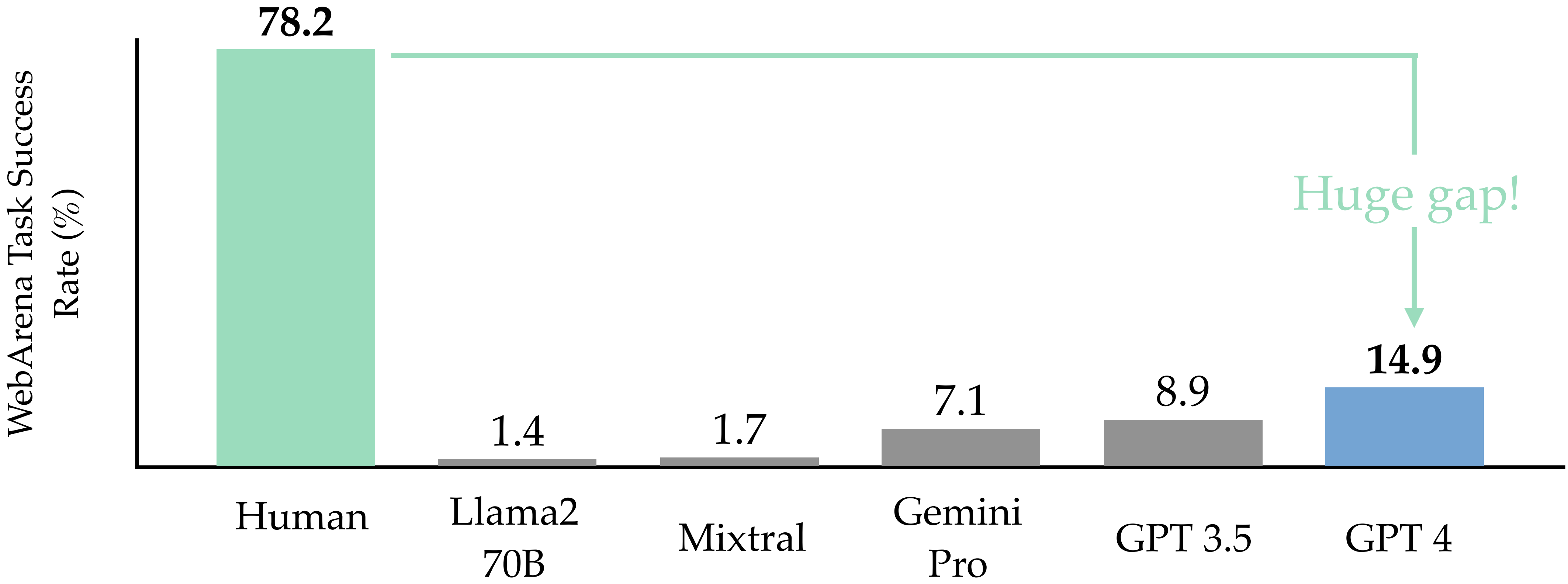
Order Information

Shipping Address

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

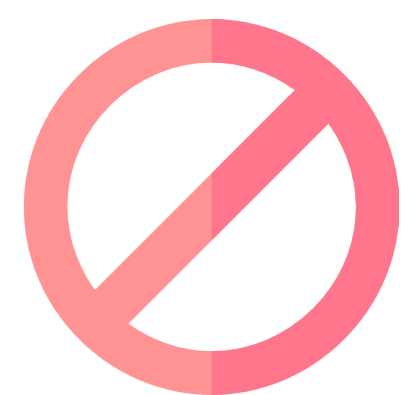
LLMs are the critical yet early step toward AI autonomy

LLMs lack several critical capabilities to be AI agents



Open-source models struggle

LLMs lack critical capabilities to be AI agents

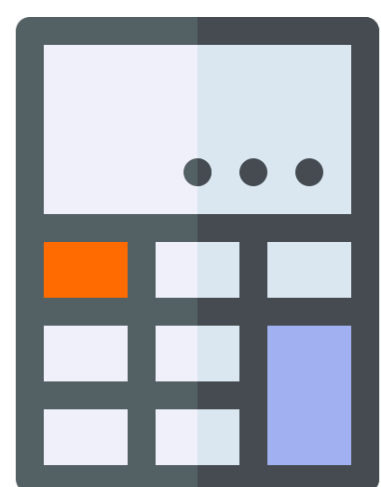


Tool use

Alex's total spend is

$78.56 \times 7 + 46.7 = 543.6$

56 days ago is $5/20/2023$



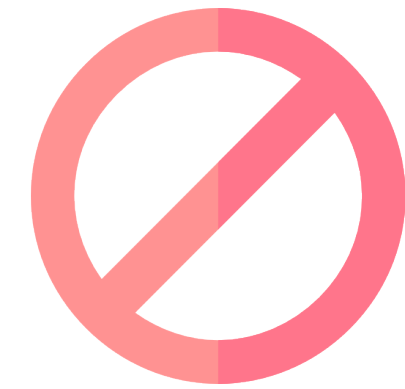
AI agents

- Employ tools to enhance accuracy and expand capabilities

LLMs

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

LLMs lack critical capabilities to be AI agents



Abstract reasoning

AI agents

- Learn the common principles
- Maintain steady and reliable performance

LLMs

- Inconsistent performance across conceptually similar tasks



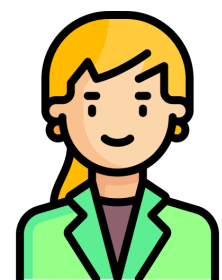
Fork `metaseq`



Fork `transformers`

Fork all repos owned by Meta

LLMs lack critical capabilities to be AI agents



Find the customer who spent [...] Send the customer [...]

Lifetime Sales
\$0.00

Average Order
\$0.00

Chart is disabled. To enable the chart, click [here](#).

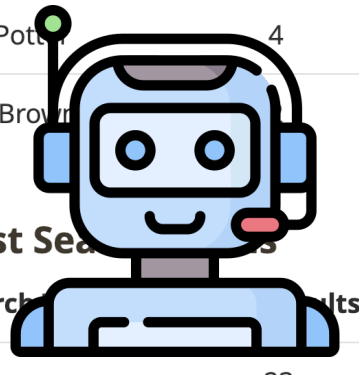
Revenue	Tax	Shipping	Quantity
\$0.00	\$0.00	\$0.00	0

Bestsellers | Most Viewed Products | New Customers | Custom

Customer	Total	Product	Price	Qty
Sara	\$94.40			
Grace	\$190.00	Sprite Stasis Ball 65 cm	\$27.00	
Matt Baker	3	Quest Lumaflex™ Band	\$19.00	
Lily Pot	4	Sprite Yoga Strap 6 foot	\$14.00	
Ava Bro	\$83.40	Overnight Duffle	\$45.00	
		Sprite Stasis Ball 55 cm	\$23.00	

Last Searches

Search	Results	Uses
tanks	23	1



How can I find all orders?

LLMs lack critical capabilities to be AI agents

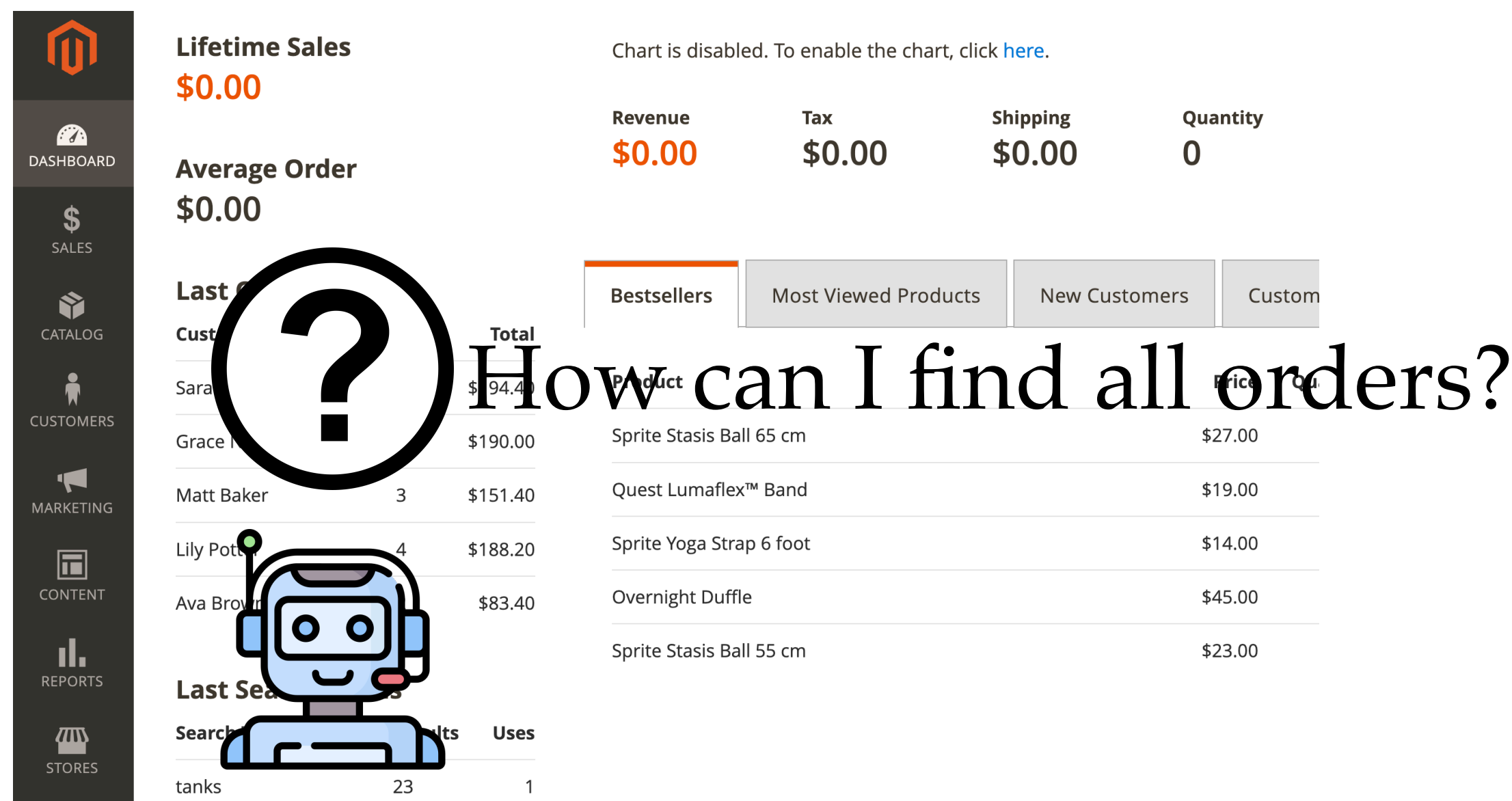
 Up-to-date knowledge

AI agents

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

LLMs

- Knowledge of LLMs is limited by the training cutoff



The screenshot shows a dashboard with various metrics and a table of products. A large question mark is overlaid on the dashboard, and a robot icon is positioned near the bottom left. The text "How can I find all orders?" is overlaid on the dashboard.

How can I find all orders?

Revenue	Tax	Shipping	Quantity
\$0.00	\$0.00	\$0.00	0

Product	Price	Qty
Sprite Stasis Ball 65 cm	\$27.00	
Quest Lumaflex™ Band	\$19.00	
Sprite Yoga Strap 6 foot	\$14.00	
Overnight Duffel	\$45.00	
Sprite Stasis Ball 55 cm	\$23.00	

GPT-4 knowledge cutoff: Sep 2021

WebArena application version: Jan 2023

Tool use
Abstract reasoning



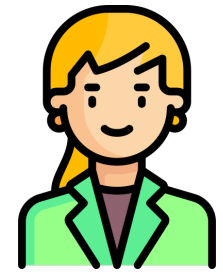
Speaking AI's
"language"

Up-to-date knowledge



Learning by
reading docs

Generating natural language for various tasks

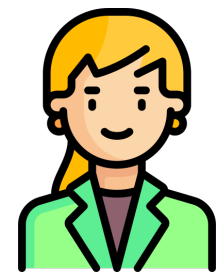


Alex Martin made three orders: \$47.51 on 9/18/2023, \$765.8 on 1/1/2024 and \$35.4 on 1/9/2024. How much he spent in my store in the last 56 days?

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
[...] $765.8 + 35.4 = \mathbf{\$785.4}$

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

Natural language exhibits limitations in performing tasks



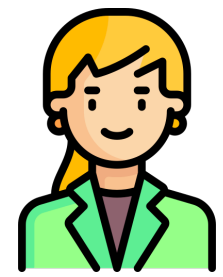
Today is 1 / 20 / 2024, Alex made three orders: \$47.51 on 9 / 18 / 2023, \$765.8 on 1 / 1 / 2024, \$35.4 on 1 / 9 / 2024. How much has he spent in the last 56 days?

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]

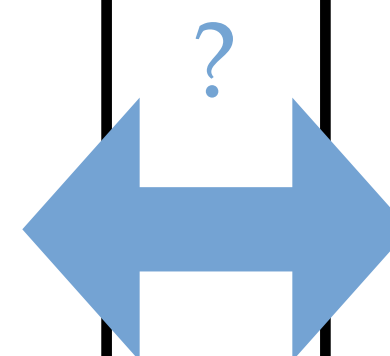
 **Confine reasoning and solving within LLMs**

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
[...] $765.8 + 35.4 = \$785.4$



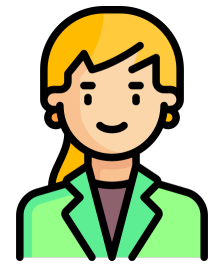
Today is **2 / 13 / 2024**. I first subtract **13** days [...] The date **192** days ago is **8 / 5 / 2023**.
[...] Order 1 was placed on 9 / 18 / 2023, which is within the last 192 days
[...] $47.51 + 765.8 + 35.4 \dots$

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

 **Confine reasoning and solving within LLMs**  **Express solutions at the example level**

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 →

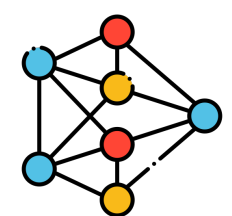
It was made on 9/18/2023 →

[...]

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)
```

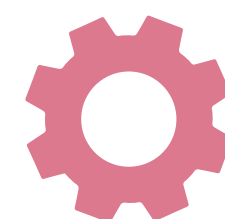
[...]

```
# check if order 1 is within the period
```

```
if order_1_date > start_date:
```

```
    alex_total_spend += order1_amount
```

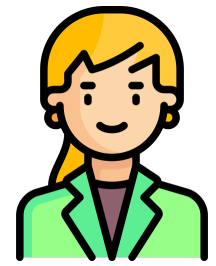
[...]



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

PaL

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



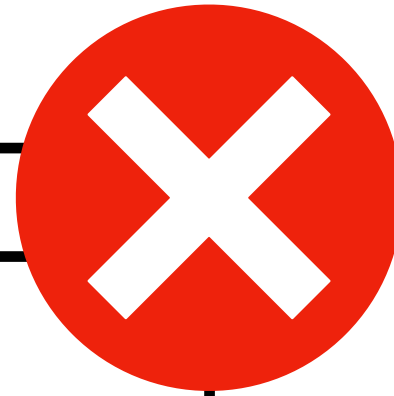
Interleave between natural language and programming language

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



- Abundant
- Easily comprehensible

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



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

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

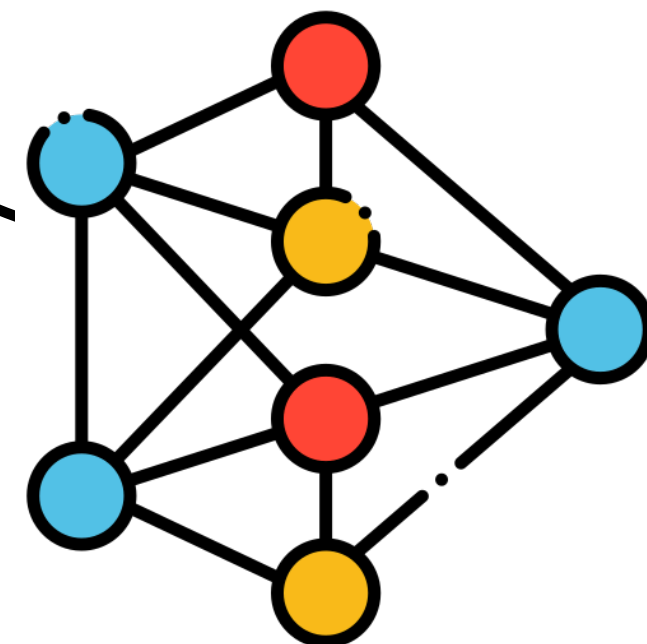
Program 2

...

In-context examples

- Manually create
- Select from a training set

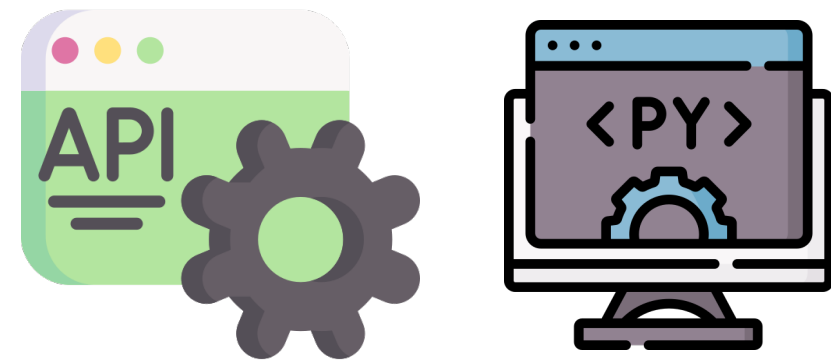
coding-proficient LLM



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



PaL offloads the solving to tools seamlessly

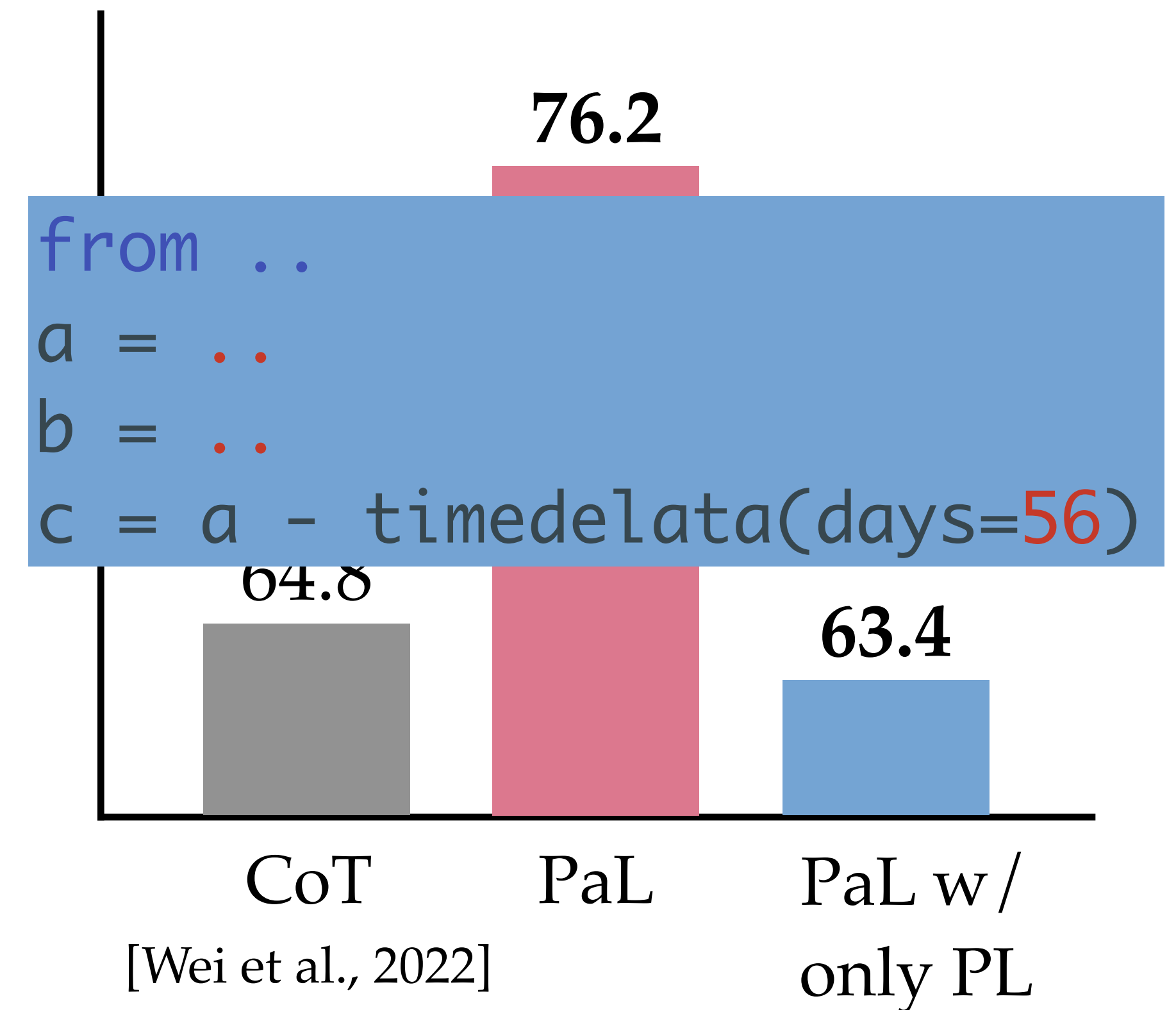


Today is 1/20/2024 [...] How much has he spent in the last 56 days?

```
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)



[Chowdhery et al, PaLM]

[Mishra et al, Lila]

[Austin et al, Learning ..]

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?

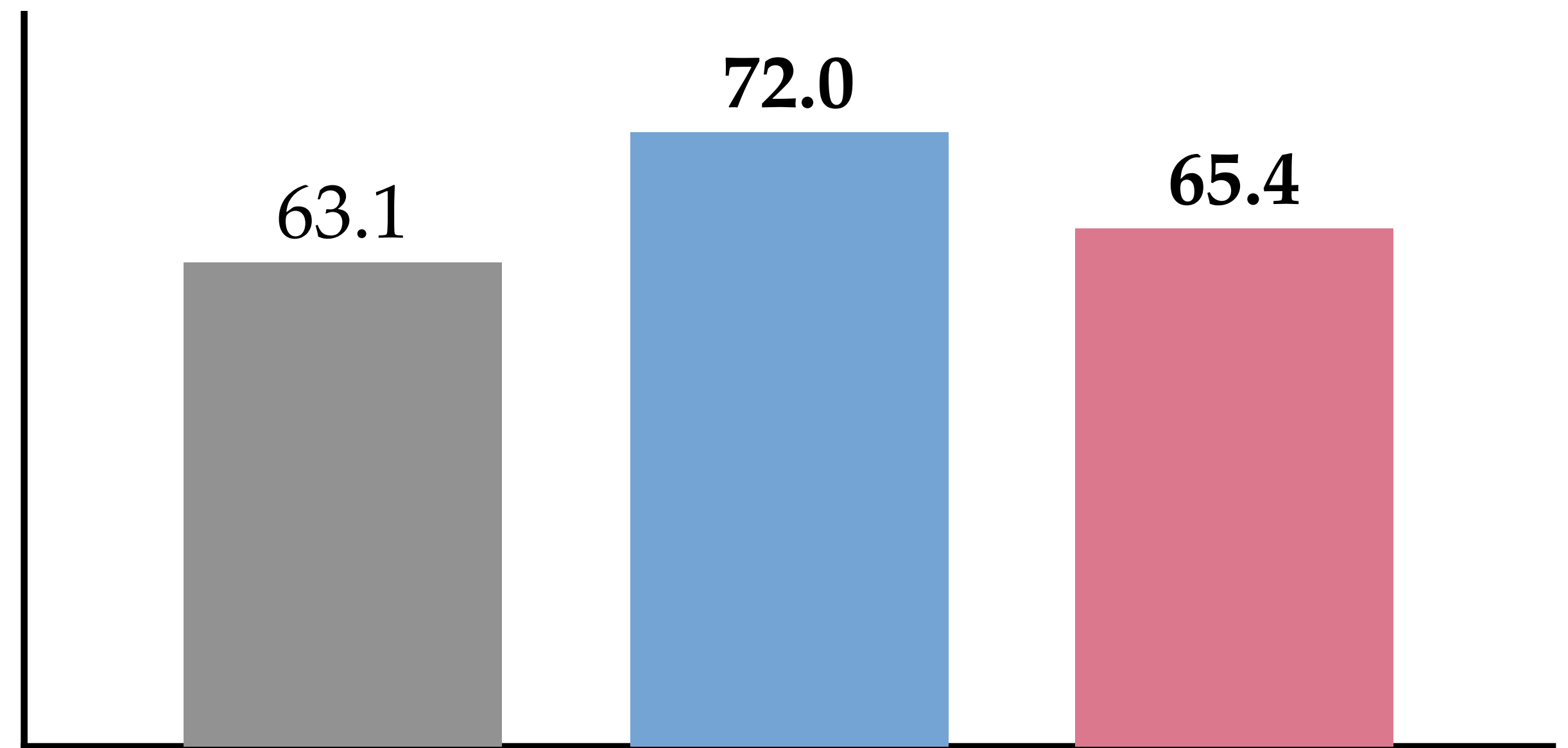
[...] the total of two orders is
 $765.8 + 35.8$ [...]

`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

Task solving accuracy (%) on GSM8k



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
[...] $47.51 + 765.8 + 35.4 \dots$

Indirect

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
[...]
```

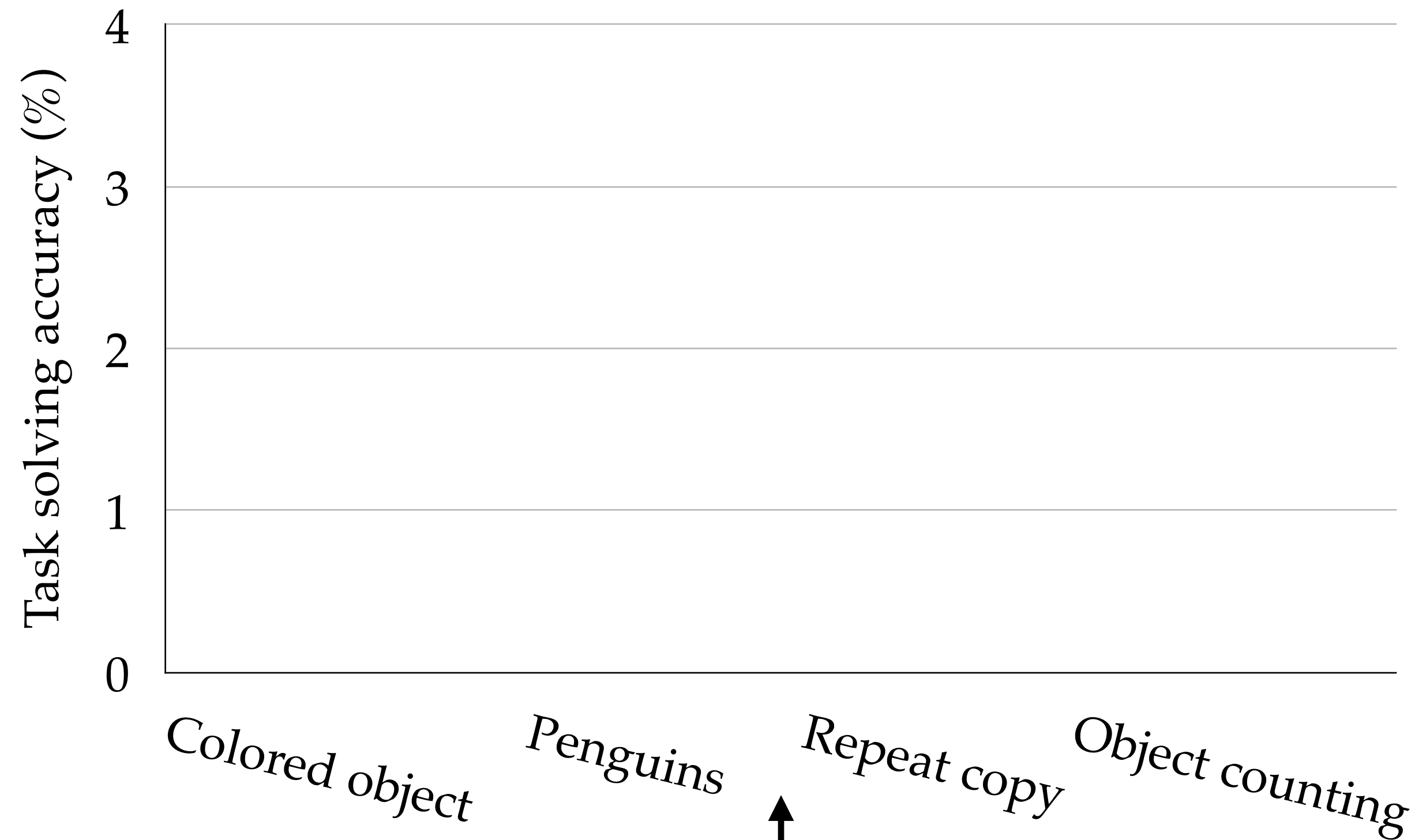
```
today = datetime(2024,2,13)
start_date = today - \
    timedelta(days=192)
[...]
if order_1_date > start_date:
    total += order_1_amount
[...]
```

direct



PaL

Programs enhance LLMs in using in-context examples

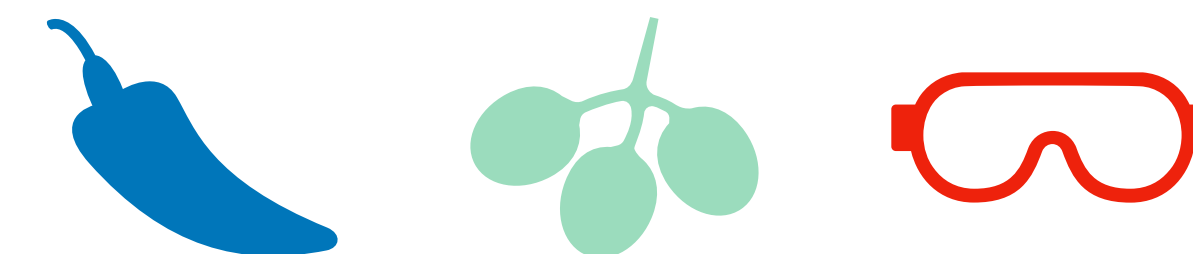


Datasets where different examples share common problem-solving strategies

- Maintain an object attribute list
- Spatial reasoning



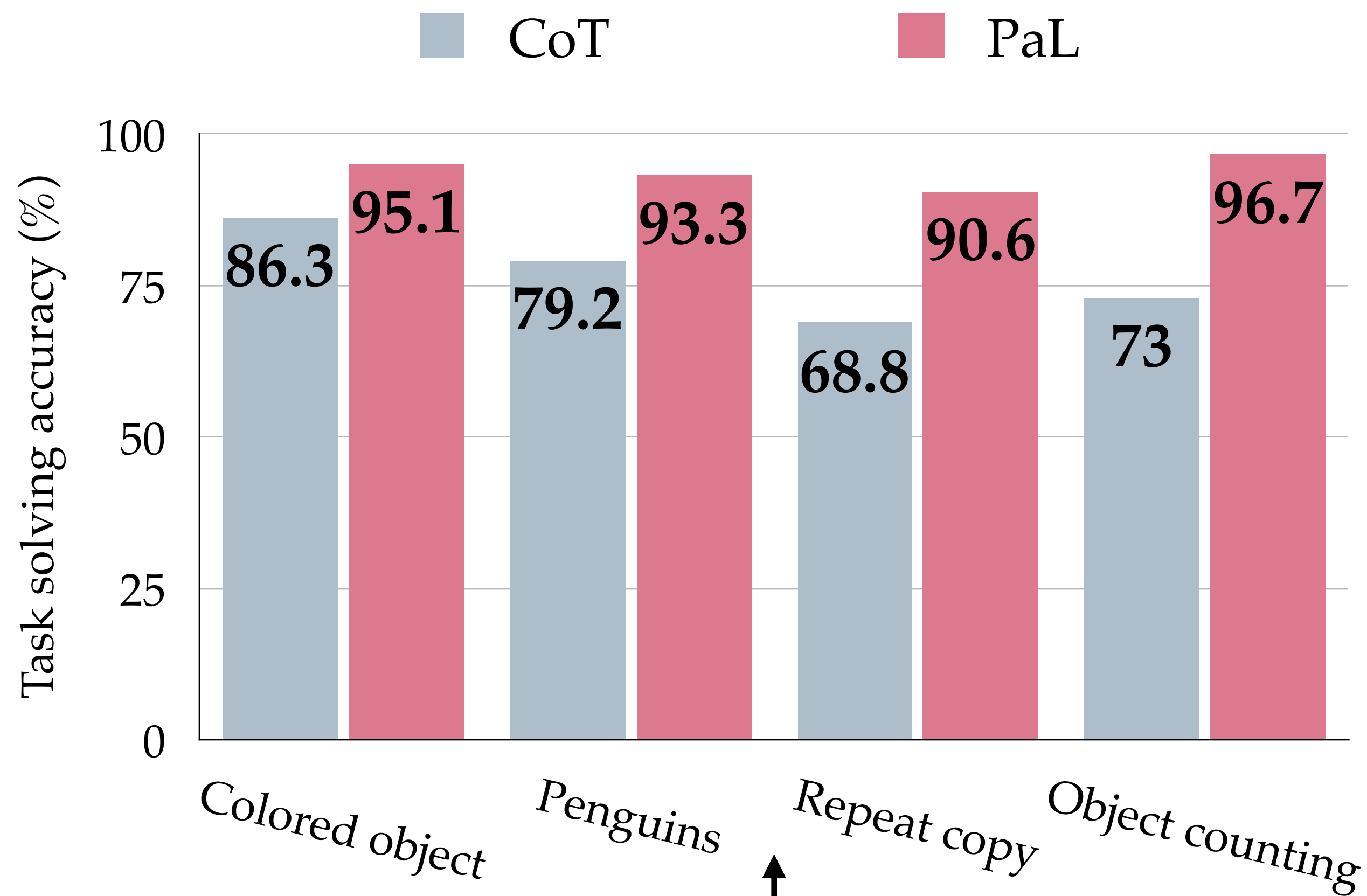
What's the color of the right most object?



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

Example tasks in colored objects

Programs enhance LLMs in using in-context examples



Datasets where different examples share common problem-solving strategies

Bonus: Programs naturally encode structures

“Get Alex’s total spend within 56 days”

class Graph:

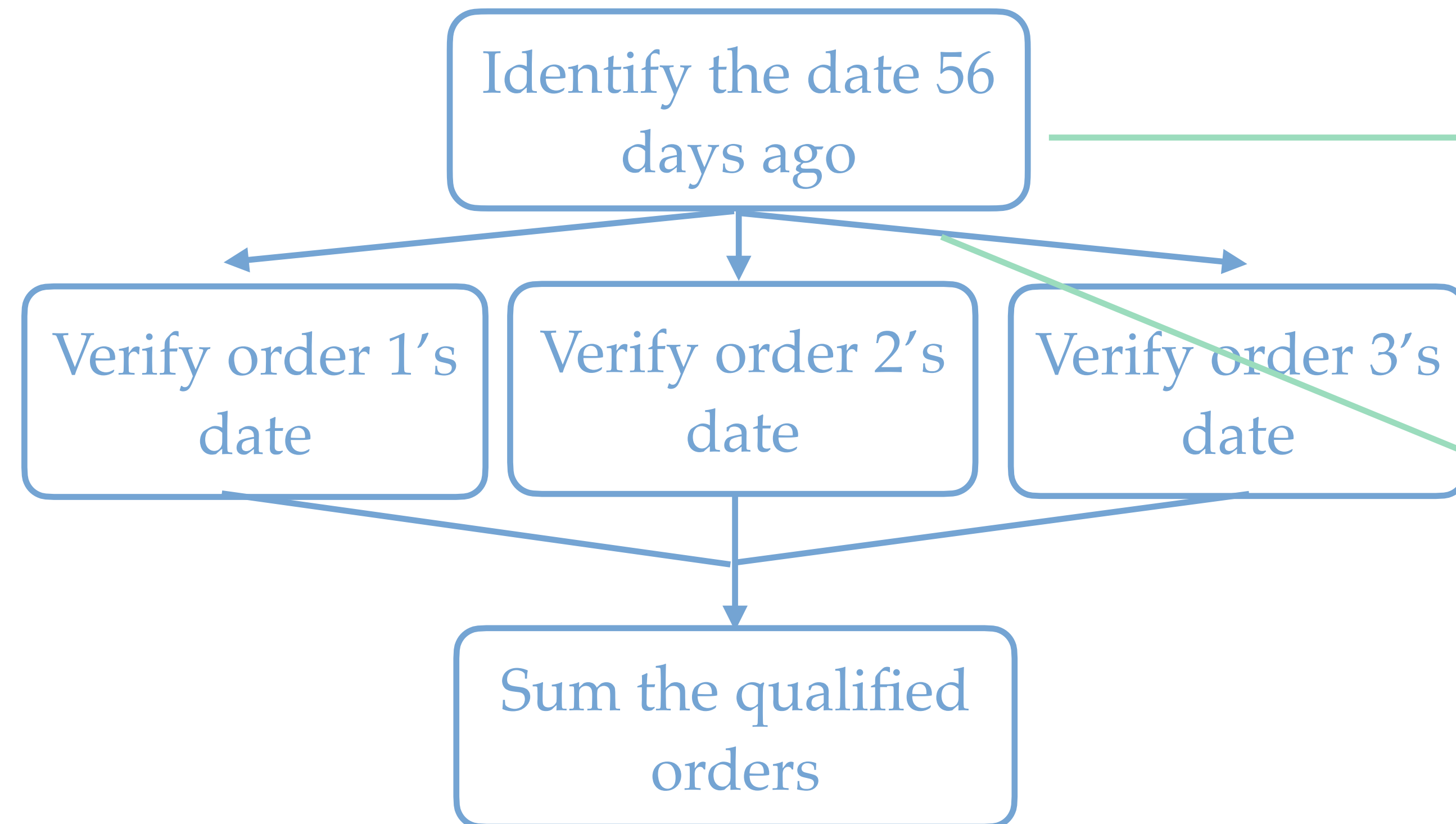
```
goal = "Get the total spend of  
Alex within 56 days"
```

```
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



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
  }
}
```


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

+ Multi-sample generation

[Zhou et al, PaL]

+ More modularized planning

[PaL, Jiang et al]

+ Execution feedback

[Wang et al, Sun et al]

+ APIs for other modalities

[Lu et al, Stanic et al]

+ Finetune with program-aided
solution for specific domains
(e.g., math)

[Yue et al, Xu et al]

Improve program
generation quality

For multi-modal
tasks

Sophisticated domain
models



PaL



Evaluating AI
agents



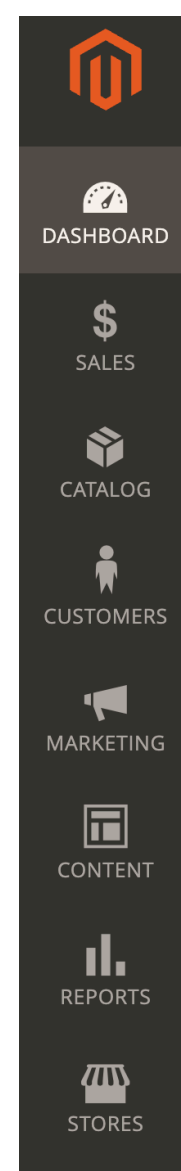
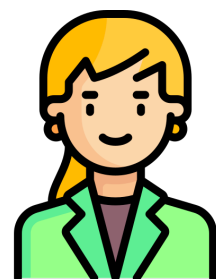
Speaking AI's
"language"



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.



Lifetime Sales
\$0.00

Average Order
\$0.00

Last Orders

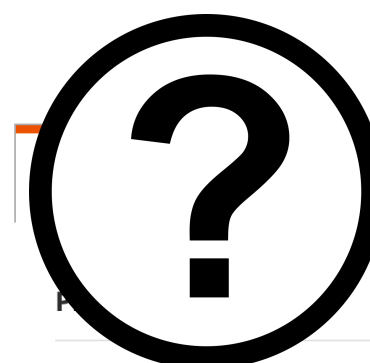
Customer	Items	Total
Sarah Miller	5	\$194.40
Grace Nguyen	4	\$190.00
Matt Baker	3	\$151.40
Lily Potter	4	\$188.20
Ava Brown	2	\$83.40

Last Search Terms

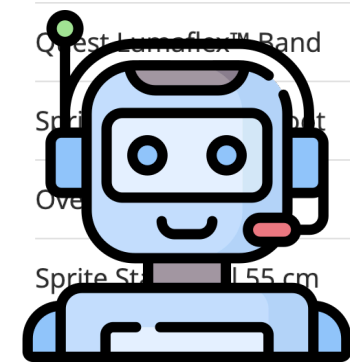
Search Term	Results	Uses
tanks	23	1

Chart is disabled. To enable the chart, click [here](#).

Revenue \$0.00 Tax \$0.00 Shipping \$0.00 Quantity 0

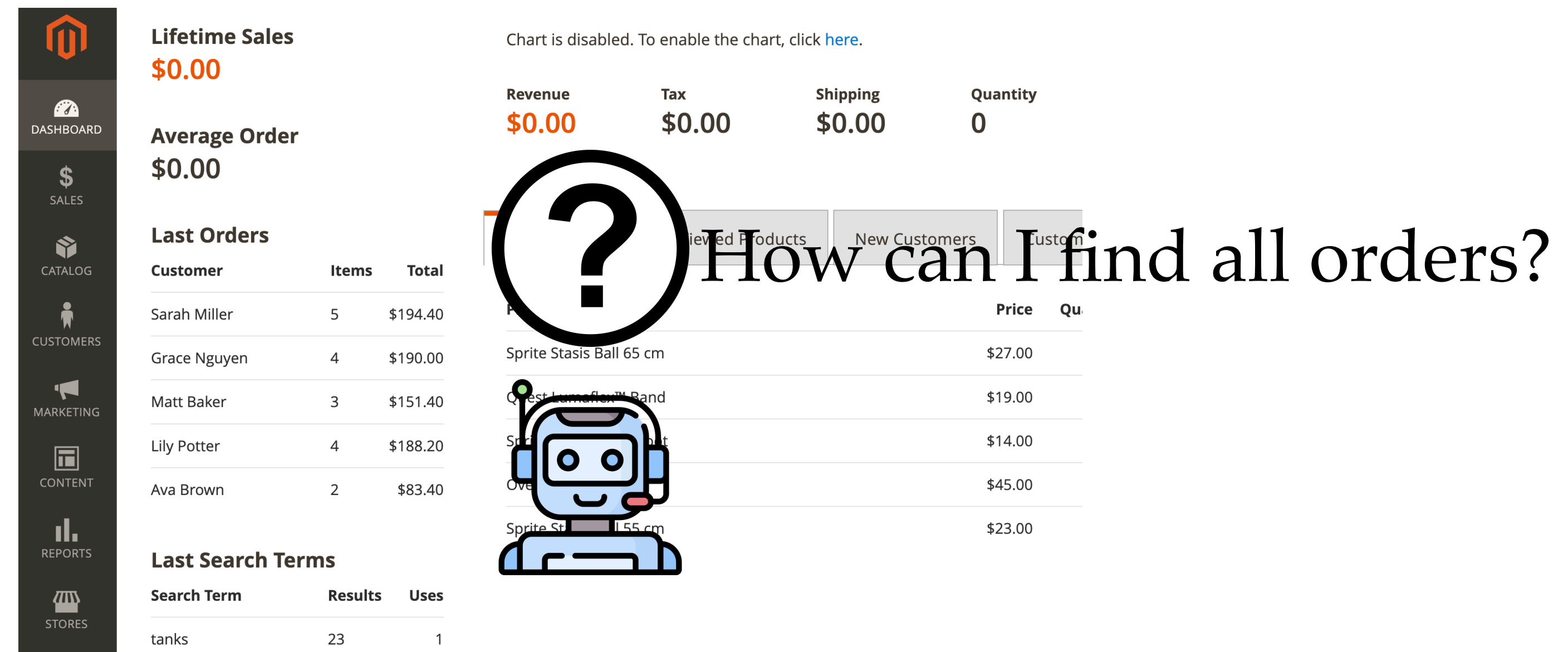


How can I find all orders?



Reviewed Products	New Customers	Custom	Price	Qu
Sprite Stasis Ball 65 cm			\$27.00	
Quest Home Alarm Band			\$19.00	
Sprite Stasis Ball 65 cm			\$14.00	
Over			\$45.00	
Sprite Stasis Ball 65 cm			\$23.00	

Knowledge is limited by the training cutoff



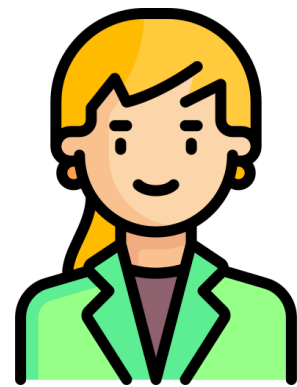
The screenshot shows a dashboard with several sections: Lifetime Sales (\$0.00), Average Order (\$0.00), Last Orders (a table with 5 rows), Last Search Terms (a table with 1 row), and a chart area that is disabled. A large question mark is overlaid on the chart area, and a robot icon is overlaid on the Last Orders table. The text "How can I find all orders?" is written across the chart area.

Customer	Items	Total
Sarah Miller	5	\$194.40
Grace Nguyen	4	\$190.00
Matt Baker	3	\$151.40
Lily Potter	4	\$188.20
Ava Brown	2	\$83.40

Search Term	Results	Uses
tanks	23	1



Humans adapt to new knowledge via reading



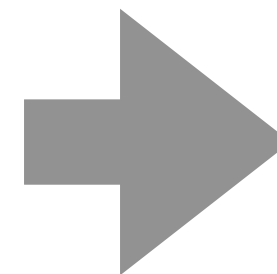
Magento OMS Docs | Getting Started | **User Guides** | Integration Guides | Features and Processes | Specifications

OMS User Guides

This User Guides section of the Order Management System (OMS) documentation provides guides to assist you in using the Magento OMS Admin.

The OMS User Guides contain the following content areas:

SECTION	DESCRIPTION	LINK
Dashboard	This section contains an overview of the Dashboard tab, a visual display of the most important information (quick search, last activity, and summaries), consolidated on a single screen for at-a-glance monitoring.	See the Dashboard user guides
Customer Service	This section details specifics of the Customer Service tab, where all customer service agents and supervisors have access to the different functionalities, such as creating returns or appeasements (which is managed through the Permissions tab).	See the Customer Service user guides
Products	This section covers the Catalog and Inventory views in the Products tab, which allows users to track items and stock movements.	See the Catalog user guide See the Inventory user guide
System	This section contains information about the Fulfillment, Permissions, Tools, Events, and Other Settings views in the System tab, and all you can accomplish in those areas.	See the System user guides
Sales	This section details all the operations that users can initiate from the Operations and Reports views in the Sales tab.	See the Sales user guides
SI Portal	This section details the various configuration areas in the SI Portal and how to access, search, and use the portal.	See the SI Portal user guides



Dashboard

Scope: All Store Views ? Reload Data

It's time to change your password.

Advanced Reporting

Gain new insights and take command of your business' performance, using our dynamic product, order, and customer reports tailored to your customer data. Go to Advanced Reporting

Lifetime Sales
\$0.00

Chart is disabled. To enable the chart, click [here](#).

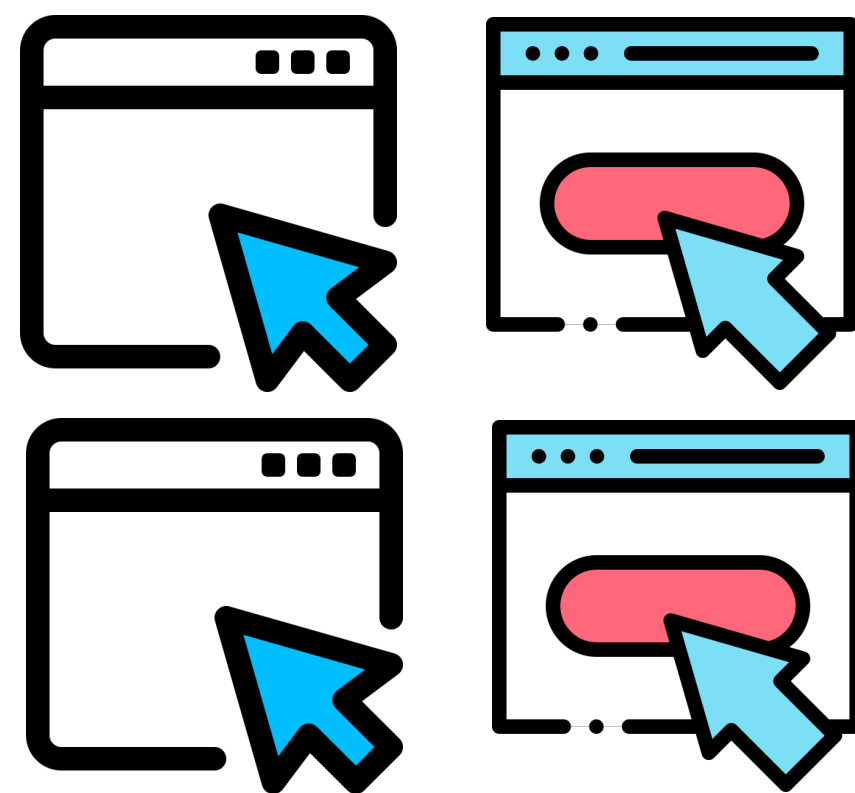
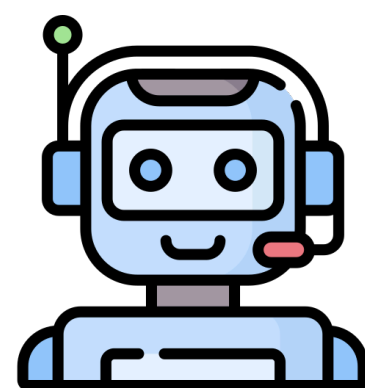
Revenue	Tax	Shipping	Quantity
\$0.00	\$0.00	\$0.00	0

Average Order
\$0.00

Last Orders

Customer	Items	Total
Sarah Miller	5	\$194.40
Grace Nguyen	4	\$190.00
Matt Baker	3	\$151.40
Lilv Potter	4	\$188.20

Product	Price	Quantity
Bestsellers		
Most Viewed Products		
New Customers		
Customers		
Sprite Stasis Ball 65 cm	\$27.00	6
Quest Lumaflex™ Band	\$19.00	6
Sprite Yoga Strap 6 foot	\$14.00	6



Direct demonstrations

Not available for new knowledge

Study scenario: using new tools by reading tool docs



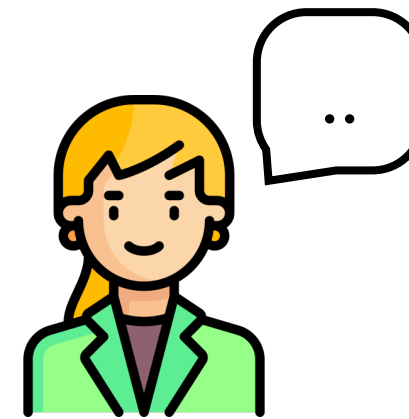
Bash commands

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.

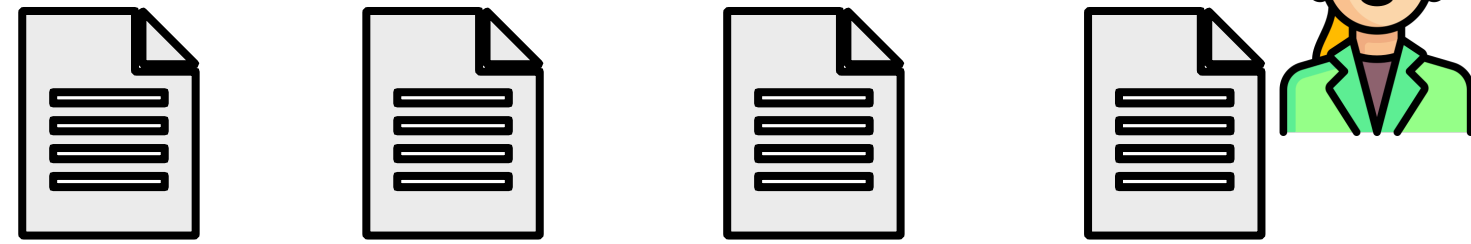
The `prefix`, `suffix`, and `dir` arguments are the same as for `mkstemp()`.

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

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

DocPrompting: Retrieval-then-generation

Docs for new commands



View slurm jobs submitted by John

queue is used to view job
... by Slurm.

queue is used to view job
... by Slurm

-u <user_list> —user=<..
Specify the usernames ...

-u <user_list> —user=<..
Specify the usernames ...

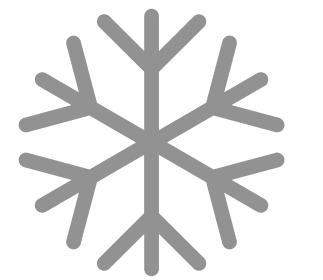
queue -u john

-i <seconds>, -- ...

Retriever

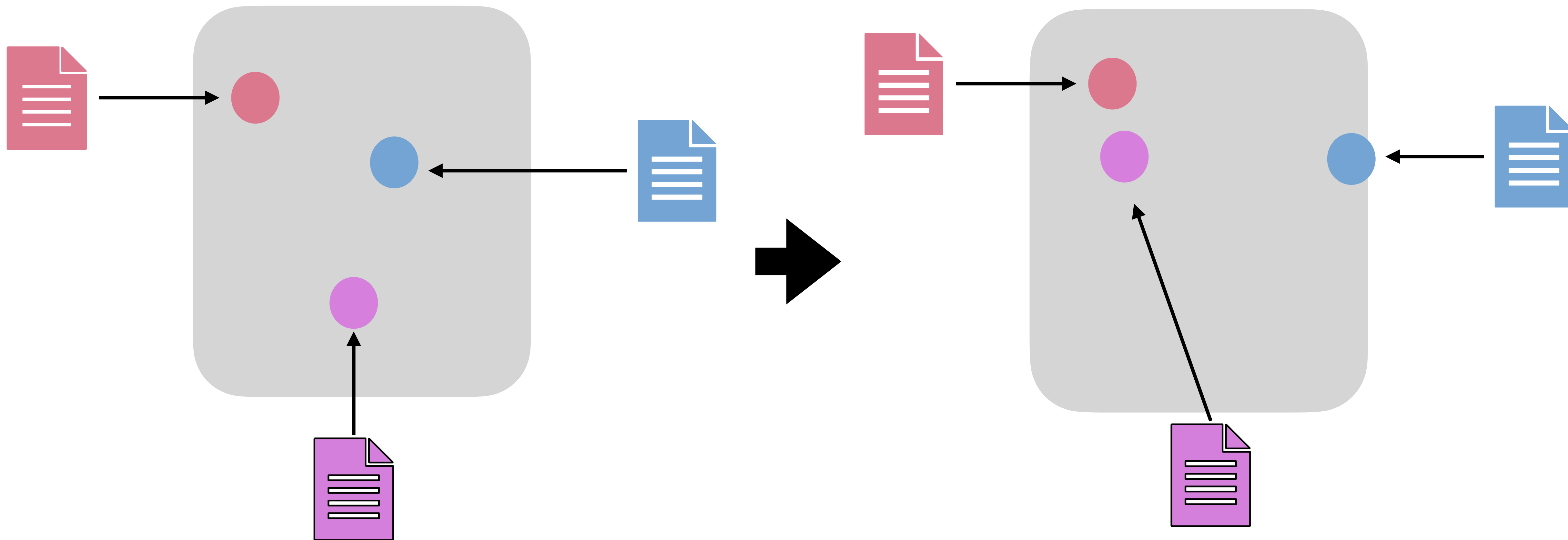
Generator

-j, <job_id_list> ...



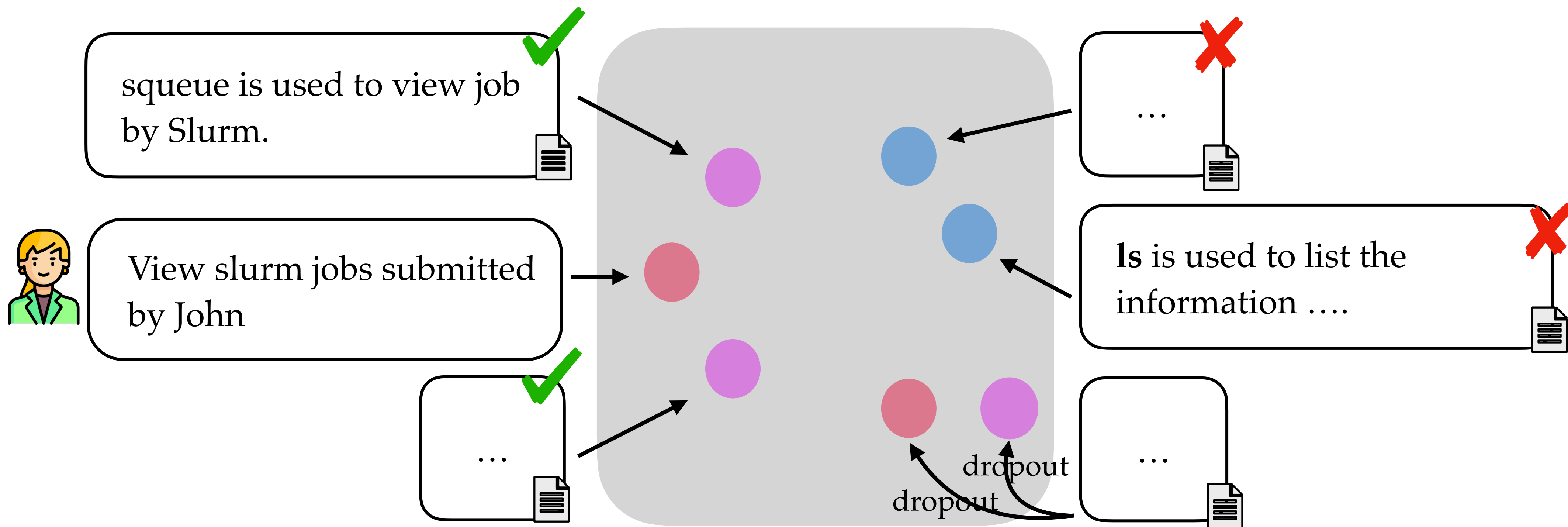
Contrastively training the doc retriever

$$\mathcal{L}^r = -\log \frac{\exp(\text{sim}(\text{red circle}, \text{purple circle})) \text{ Cosine similarity}}{\exp(\text{sim}(\text{red circle}, \text{purple circle})) + \sum_{d_j^- \in \mathcal{B}/\mathcal{D}_n^*} \exp(\text{sim}(\text{red circle}, \text{blue circle}))}$$

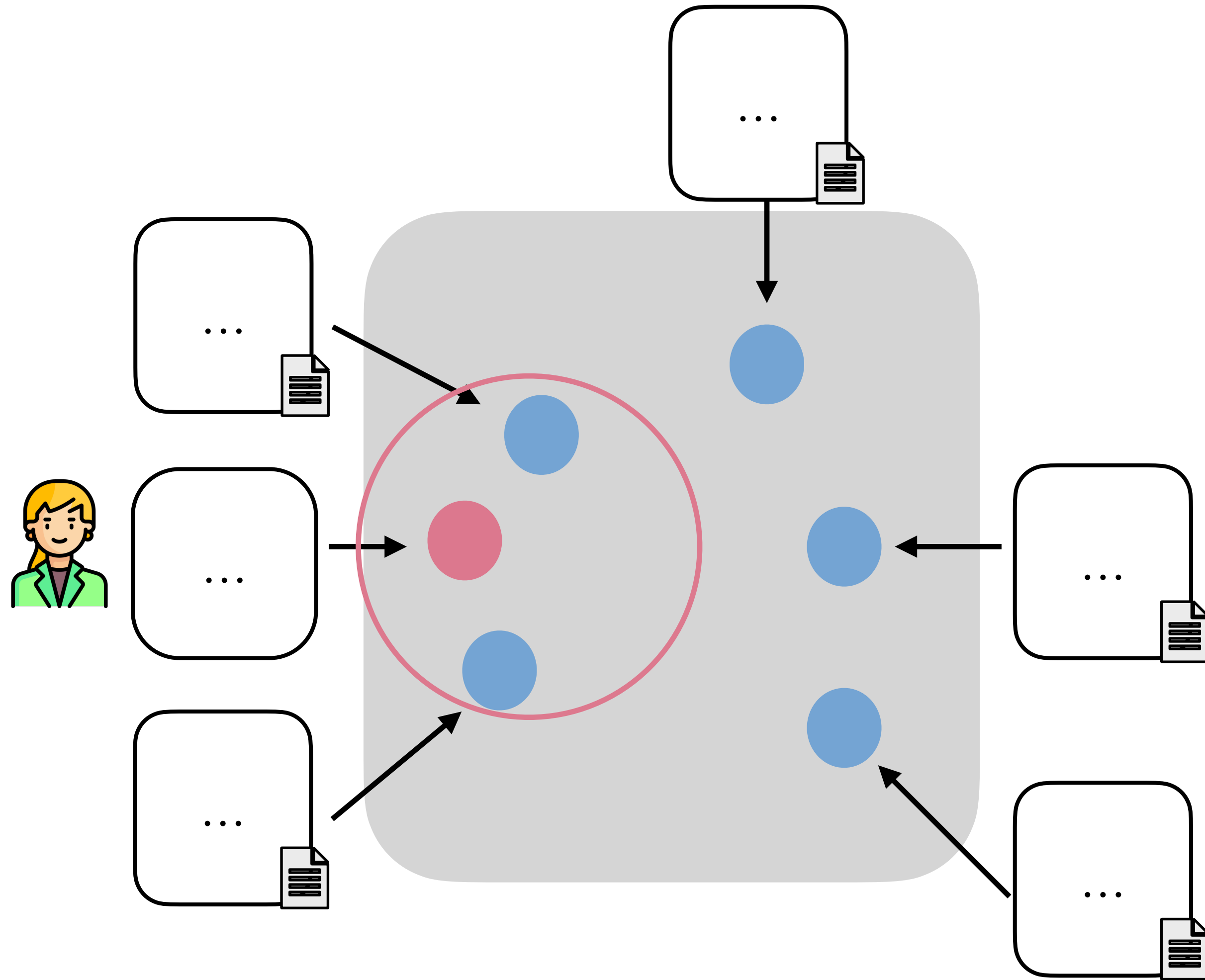


Contrastively training the doc retriever

$$\mathcal{L}^r = -\log \frac{\exp(\text{sim}(\text{red}, \text{purple})) \text{ Cosine similarity}}{\exp(\text{sim}(\text{red}, \text{purple})) + \sum_{d_j^- \in \mathcal{B}/\mathcal{D}_n^*} \exp(\text{sim}(\text{red}, \text{blue}))}$$

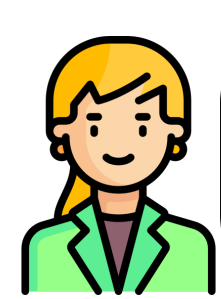


Retrieve k nearest documents



Learning to read the documents

$$\log p(c^* | \text{User} \text{ } \checkmark \text{ } \checkmark \text{ } \times \text{ })$$



View slurm jobs submitted by shuyanzh every 5 secs

Retriever retrieves irrelevant information!

queue is used to view job ... by slurm

-u <user_list> —user=<.. Specify the usernames ..

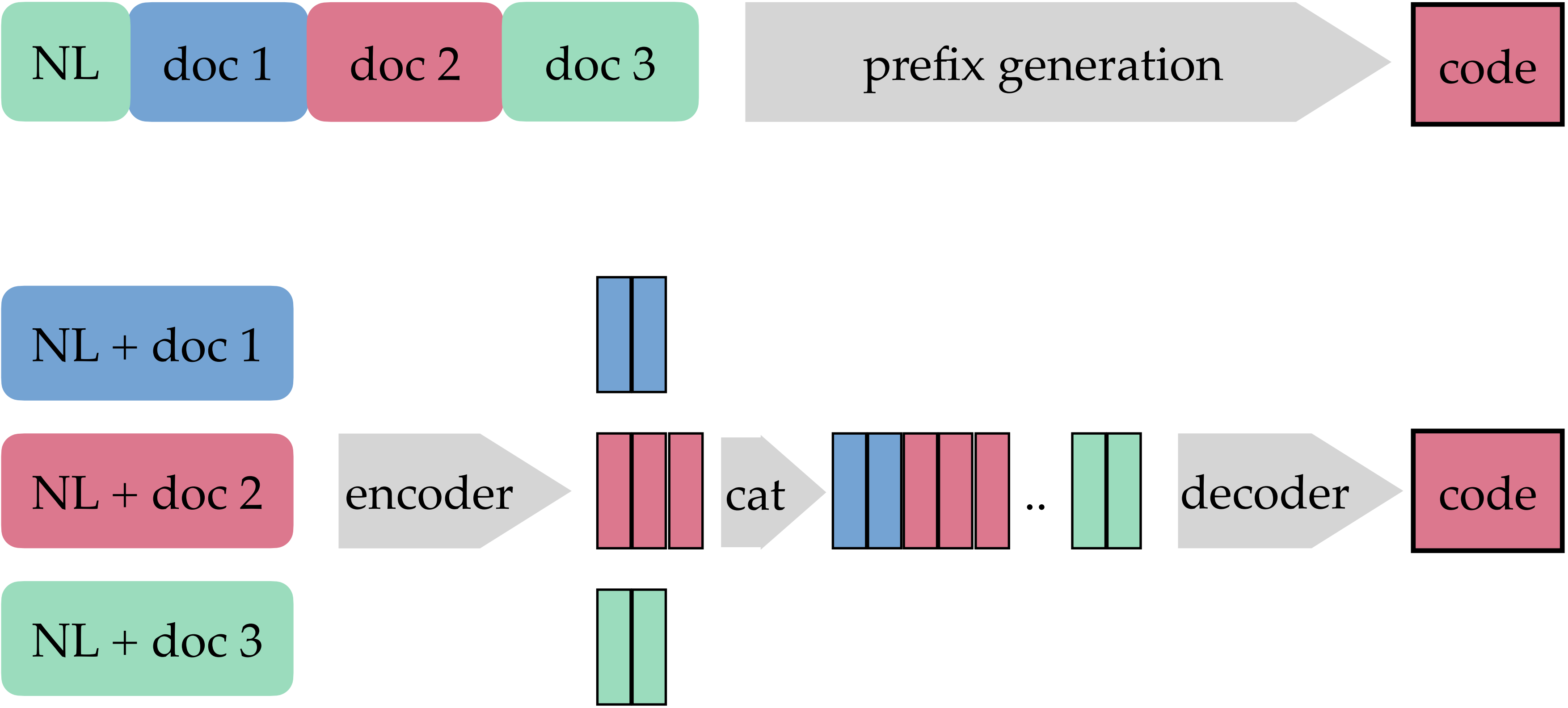
ls is used to list the information

Generator

queue -u john

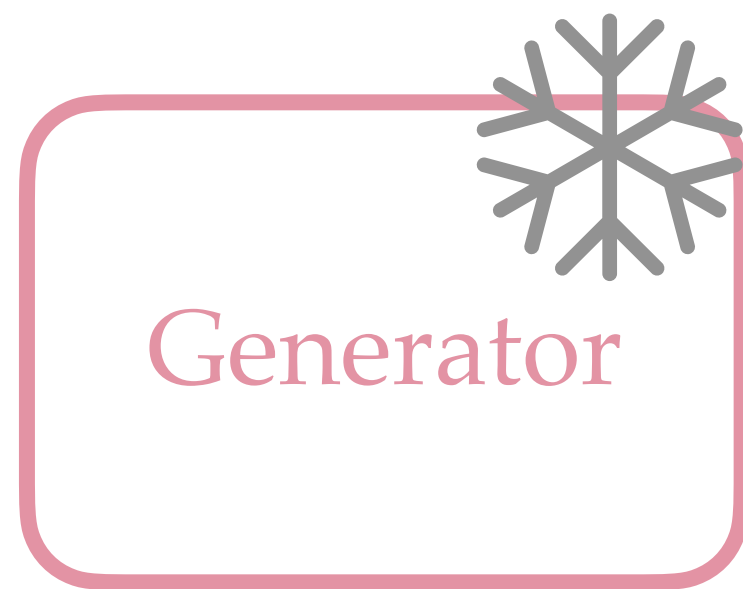
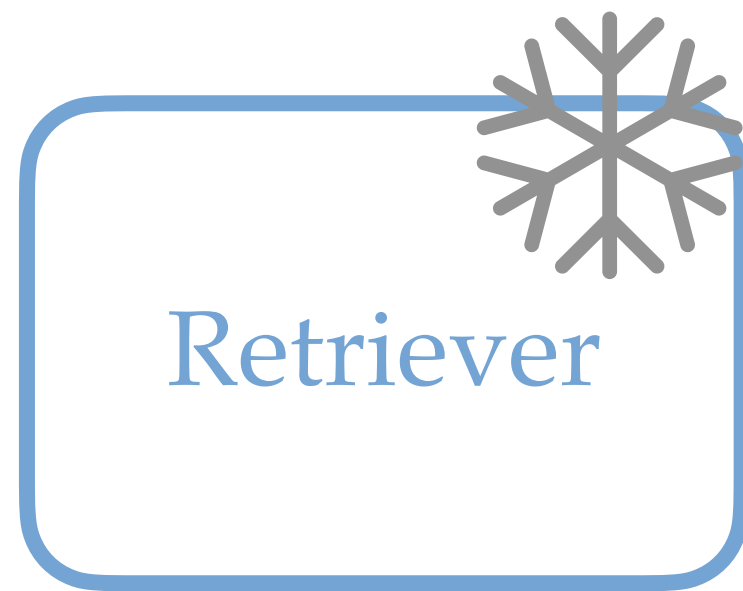
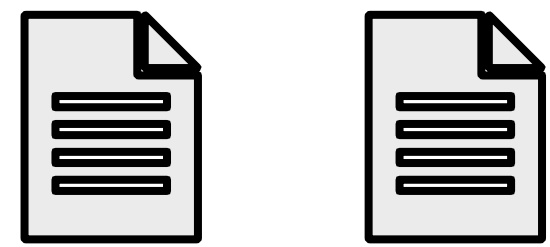
Learning to ignore irrelevant information

DocPrompting is applicable to various model architectures

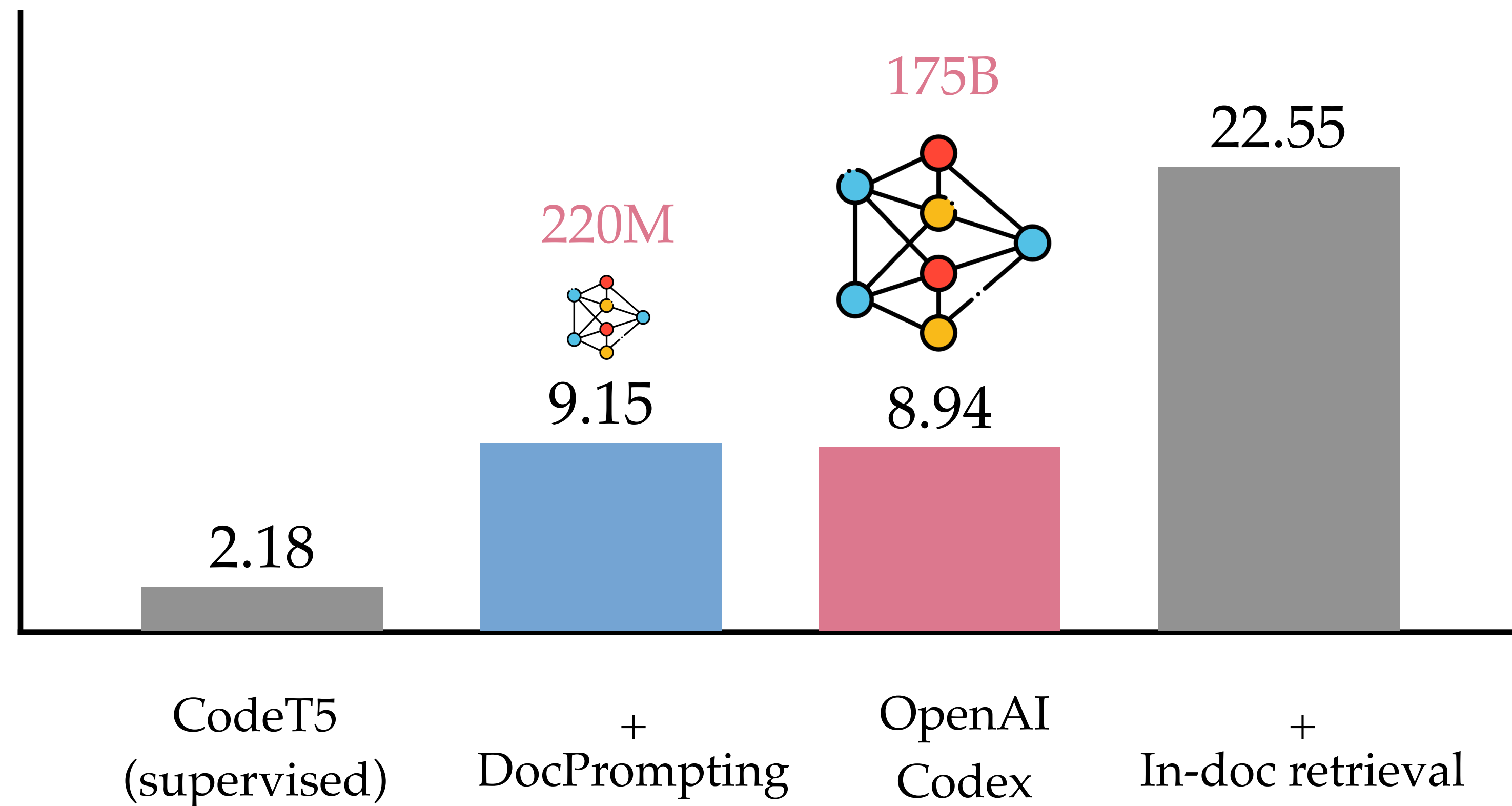


DocPrompting allows models to adapt to unseen tools without explicit demonstrations

Docs for held-out commands

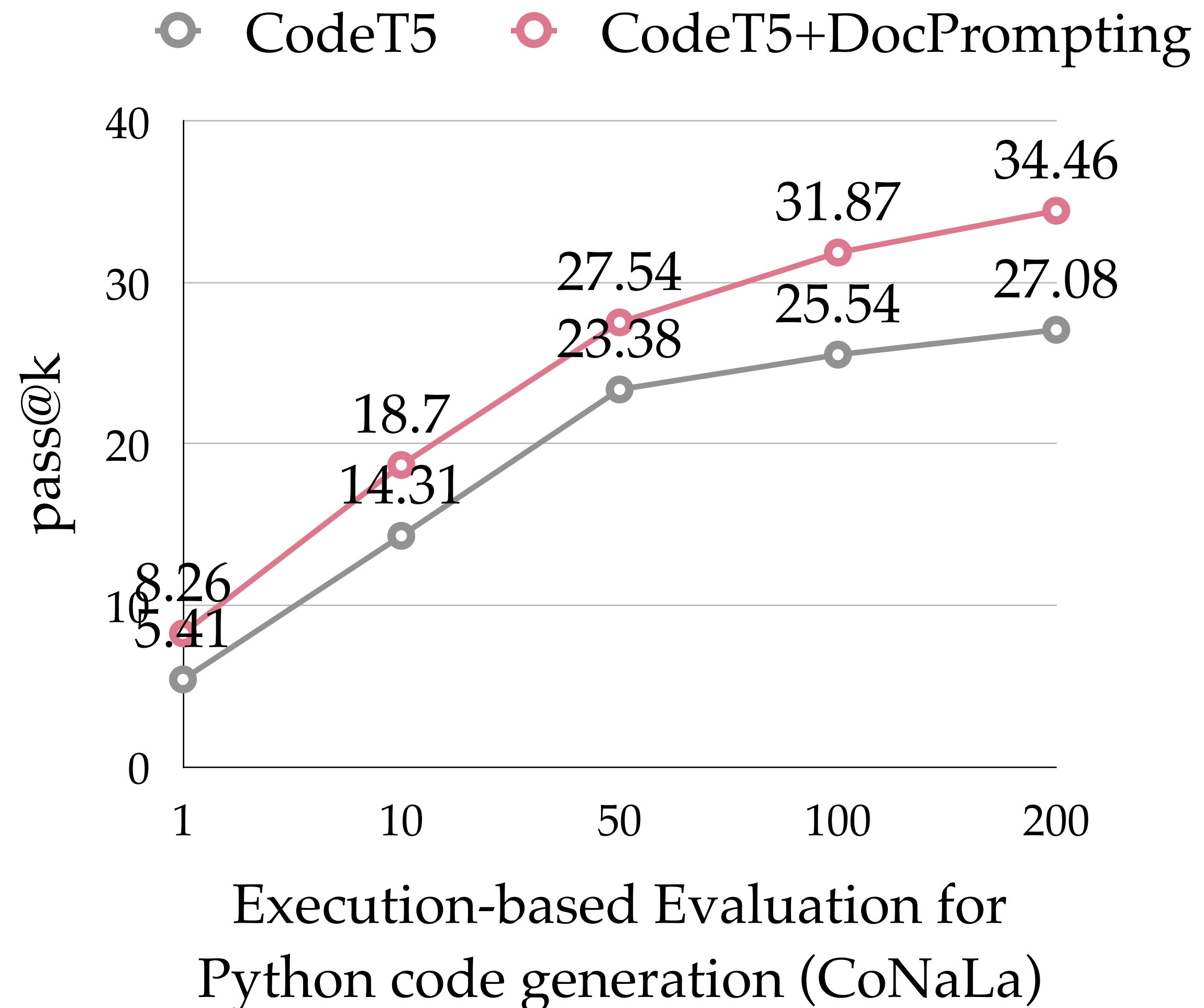
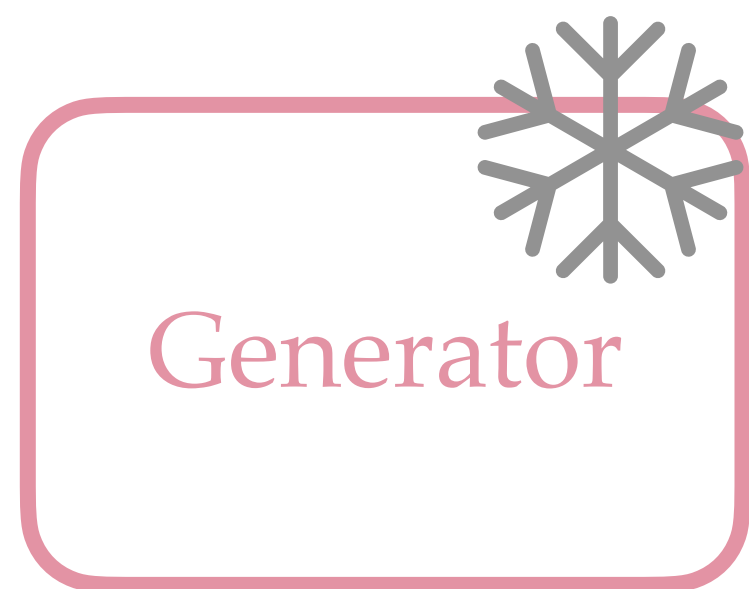
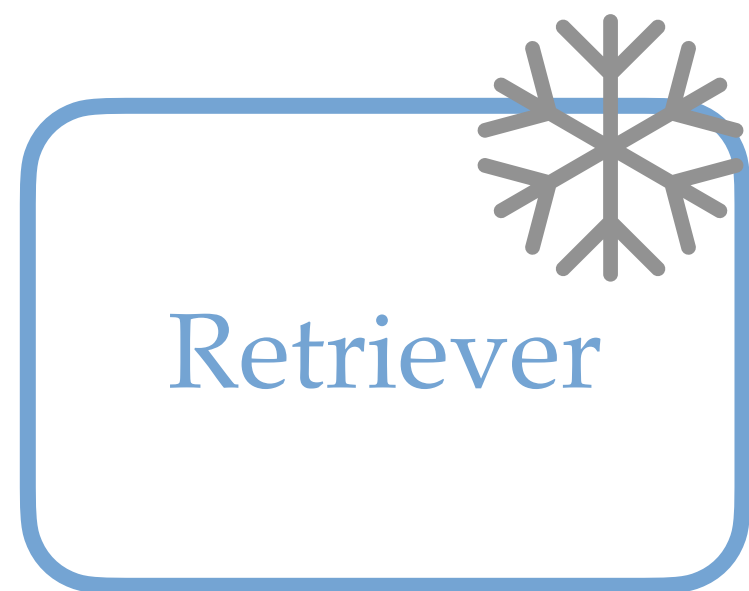
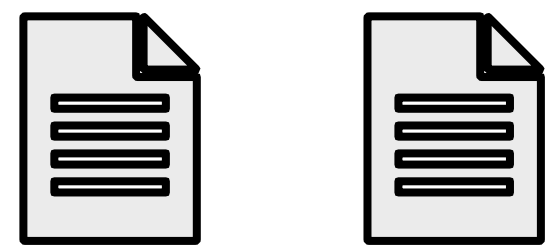


Bash command exact match (%)

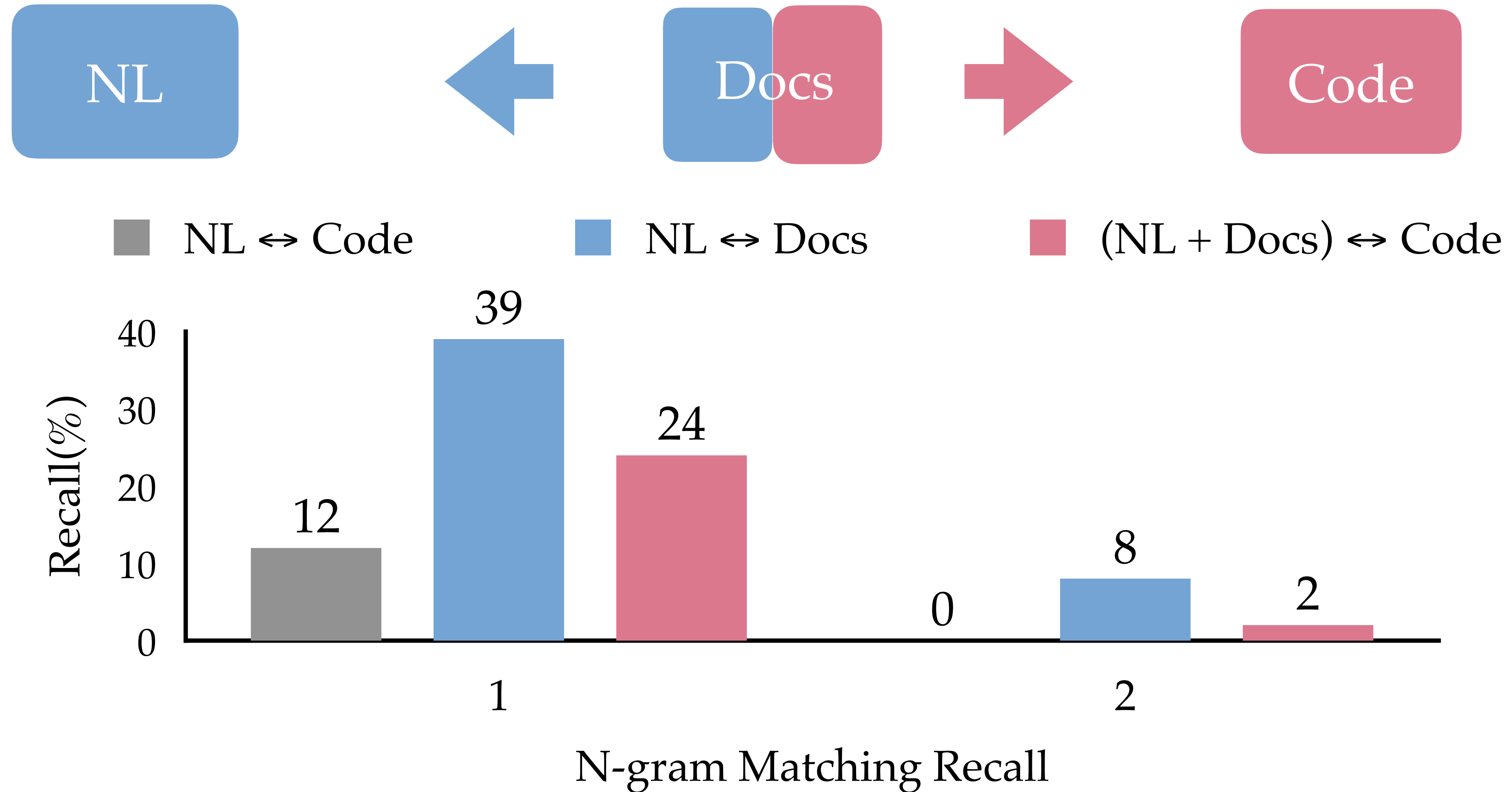


DocPrompting allows models to adapt to unseen tools without explicit demonstrations

Docs for held-out Python APIs



Docs ease the mapping between NL and code





Evaluating AI
agents

What docs created by humans that explain the tool usage

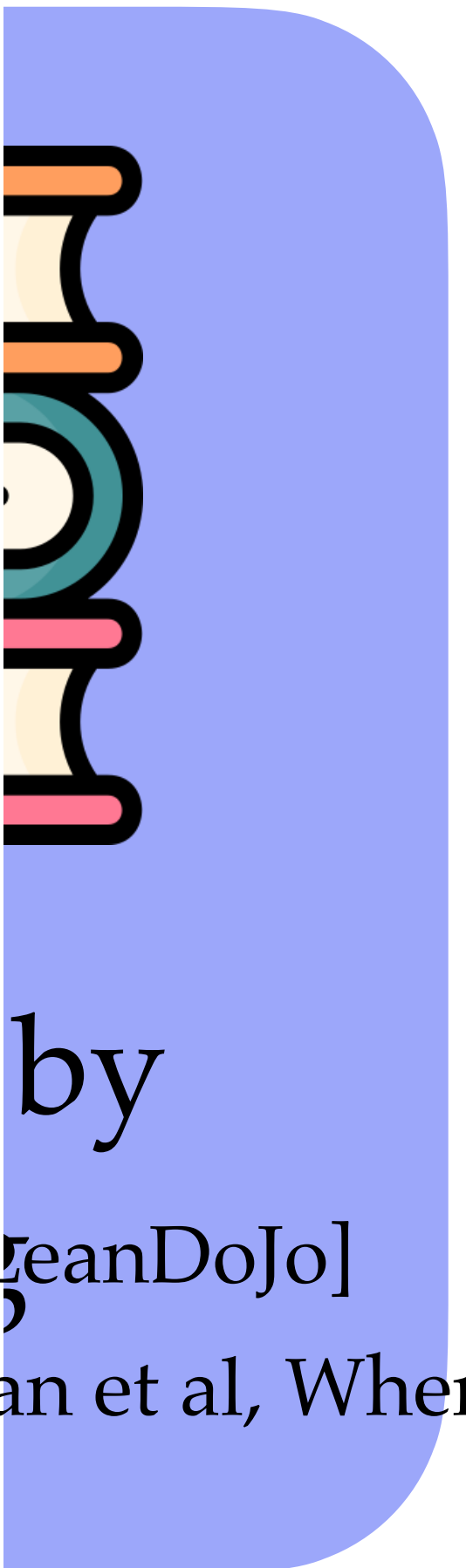
How retrieval and doc-augmented generation

✓ 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]



by