Augmented Language Models: A Survey

Authors: Meta AI

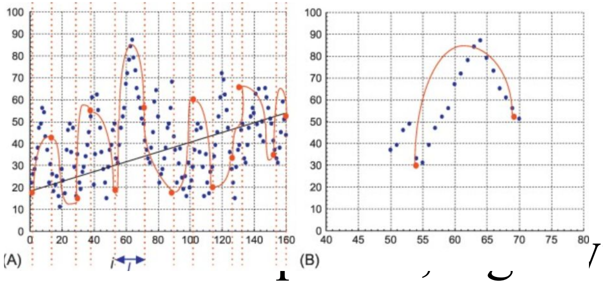
Yuer Yang (3030110653)

Department of Computer Science, The University of Hong Kong

Background

Current large language model (LLM) training issues:

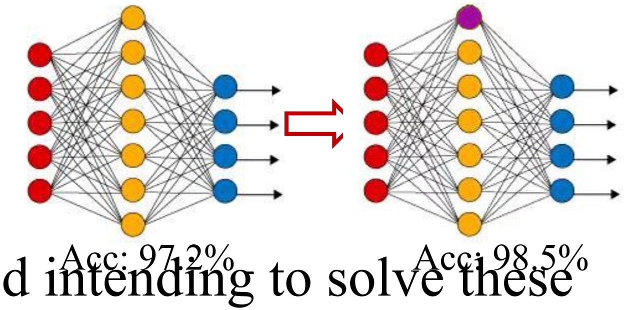
- Only given a single parametric model
- The given context is limited
- Just adjust parameters in a fixed paradigm



The next sentence is false.

It will be rainy tomorrow.

I want to graduate tomorrow.



Actually no contribution

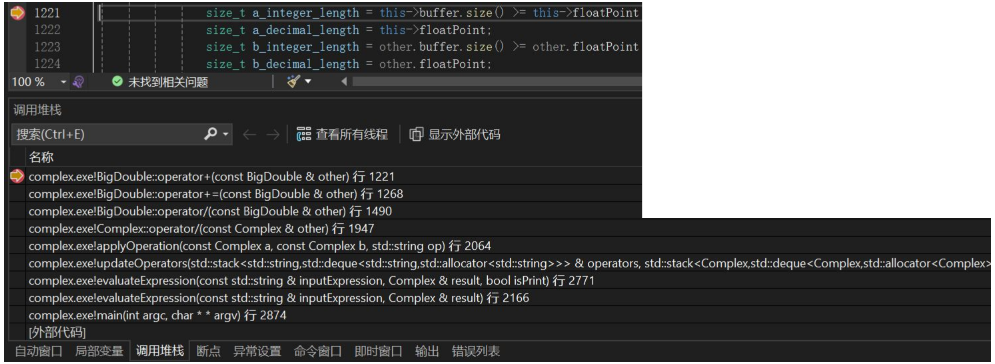
ving research trend emerged intending to solve these issues, slightly moving away from the pure statistical language modeling paradigm described above.

Overall structure

Augmented Language Mode

- Reasoning
- Tool
- Act

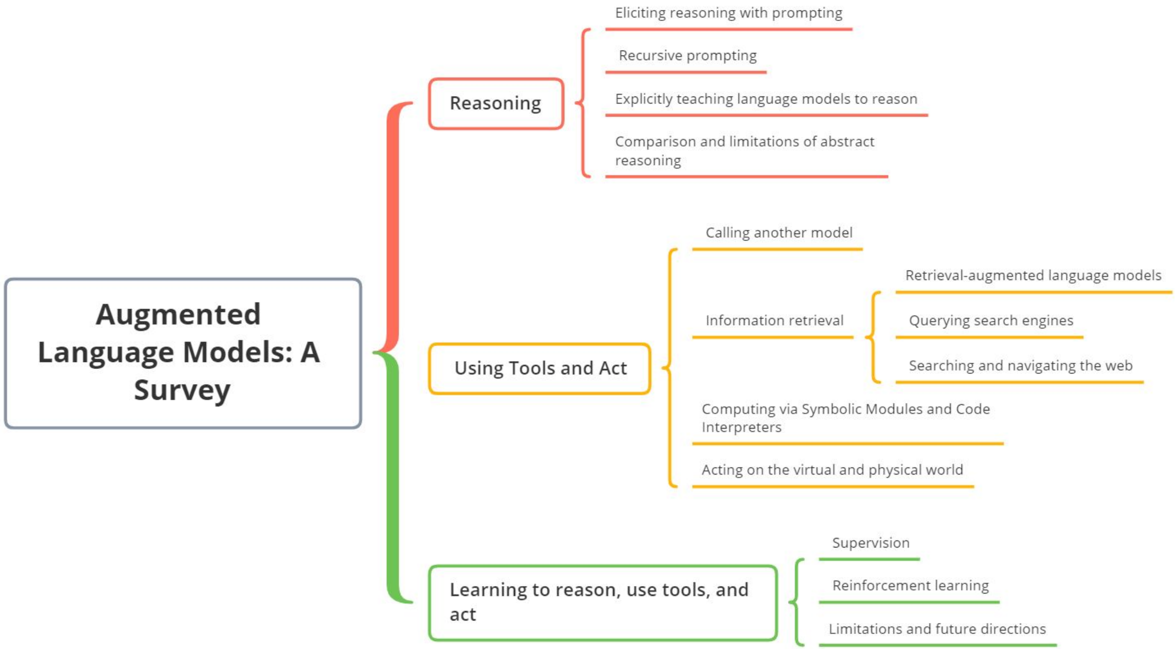
$$A = B \,\&\, B = C \rightarrow A = C$$



```
1221                size_t a_integer_length = this->buffer.size() >= this->floatPoint
1222                size_t a_decimal_length = this->floatPoint;
1223                size_t b_integer_length = other.buffer.size() >= other.floatPoint
1224                size_t b_decimal_length = other.floatPoint;
```

```
100 %   未找到相关问题

调用堆栈
搜索(Ctrl+E)   查看所有线程   显示外部代码
名称
complex.exe!BigDouble::operator+(const BigDouble & other) 行 1221
complex.exe!BigDouble::operator+=(const BigDouble & other) 行 1268
complex.exe!BigDouble::operator/(const BigDouble & other) 行 1490
complex.exe!Complex::operator/(const Complex & other) 行 1947
complex.exe!applyOperation(const Complex a, const Complex b, std::string op) 行 2064
complex.exe!updateOperators(std::stack<std::string,std::deque<std::string,std::allocator<std::string>>> & operators, std::stack<Complex,std::deque<Complex,std::allocator<Complex>
complex.exe!evaluateExpression(const std::string & inputExpression, Complex & result, bool isPrint) 行 2771
complex.exe!evaluateExpression(const std::string & inputExpression, Complex & result) 行 2166
complex.exe!main(int argc, char * * argv) 行 2874
[外部代码]
```

自动窗口
局部变量
调用堆栈
断点
异常设置
命令窗口
即时窗口
输出
错误列表

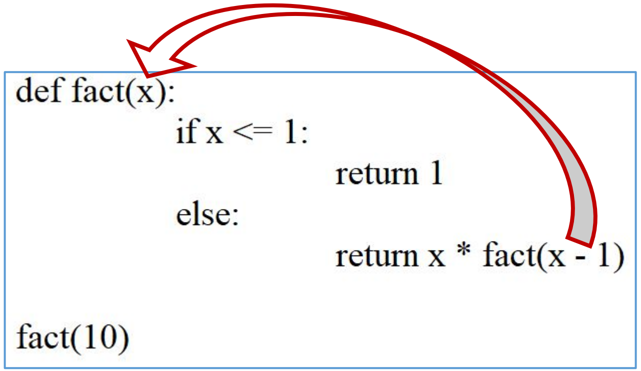


Augmented Language Models: A Survey
Reasoning
Eliciting reasoning with prompting
Recursive prompting
Explicitly teaching language models to reason
Comparison and limitations of abstract reasoning
Using Tools and Act
Calling another model
Information retrieval
Retrieval-augmented language models
Querying search engines
Searching and navigating the web
Computing via Symbolic Modules and Code Interpreters
Acting on the virtual and physical world
Learning to reason, use tools, and act
Supervision
Reinforcement learning
Limitations and future directions

Reasoning

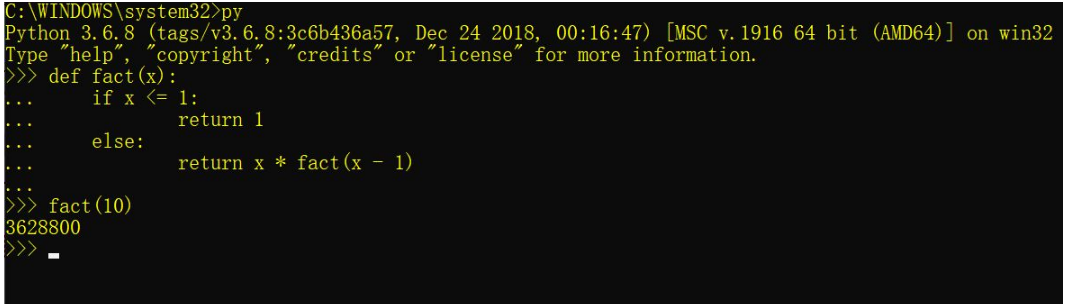
Reasoning

Reasoning is the ability to make inferences using evidence and logic. Recursion is also a very important capability.



```
def fact(x):
        if x <= 1:
                return 1
        else:
                return x * fact(x - 1)

fact(10)
```

- Eliciting reasoning with prompting
- Recursive prompting
- Explicitly teaching language models
- Abstract reasoning



```
C:\WINDOWS\system32>py
Python 3.6.8 (tags/v3.6.8:3c6b436a57, Dec 24 2018, 00:16:47) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> def fact(x):
...     if x <= 1:
...             return 1
...     else:
...             return x * fact(x - 1)
...
>>> fact(10)
3628800
>>> _
```

Reasoning - Eliciting reasoning with prompting

Question: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

Answer: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. 5 + 6 = 11. The answer is 11.

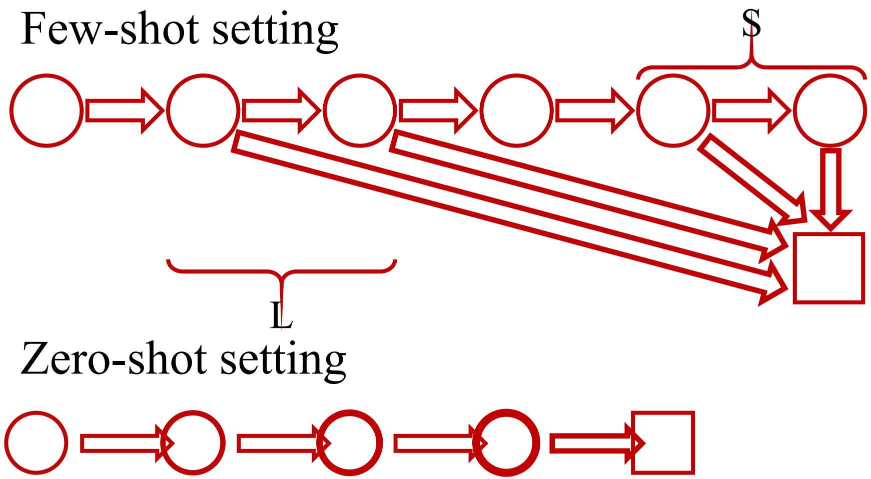
Question: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Answer: <LM>

This answer would consider not only the logic behind the second question but also **the relationship between the first and the second question**.

It performs reasoning the second question with **that relationship**.

Reasoning - Eliciting reasoning with prompting

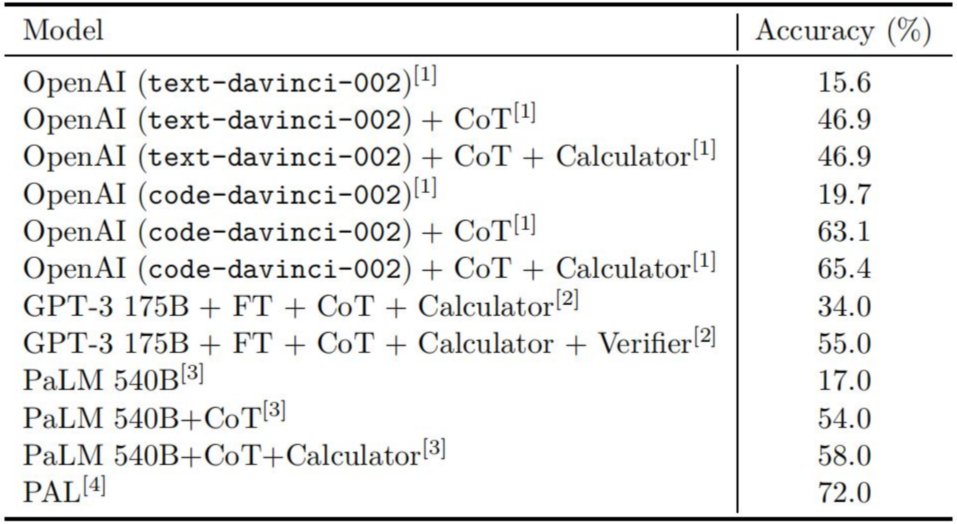


Few-shot setting

Zero-shot setting

Question: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Answer: Let’s think step by step. <LM>



Model	Accuracy (%)
OpenAI (`text-davinci-002`)[1]	15.6
OpenAI (`text-davinci-002`) + CoT[1]	46.9
OpenAI (`text-davinci-002`) + CoT + Calculator[1]	46.9
OpenAI (`code-davinci-002`)[1]	19.7
OpenAI (`code-davinci-002`) + CoT[1]	63.1
OpenAI (`code-davinci-002`) + CoT + Calculator[1]	65.4
GPT-3 175B + FT + CoT + Calculator[2]	34.0
GPT-3 175B + FT + CoT + Calculator + Verifier[2]	55.0
PaLM 540B[3]	17.0
PaLM 540B+CoT[3]	54.0
PaLM 540B+CoT+Calculator[3]	58.0
PAL[4]	72.0

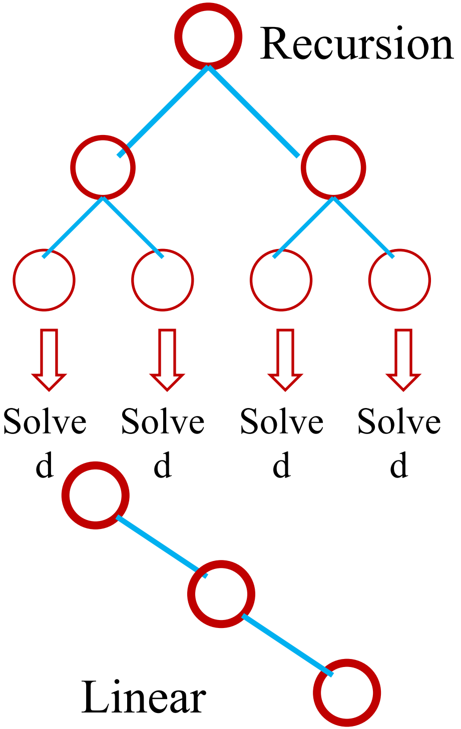
The accuracy of different models.

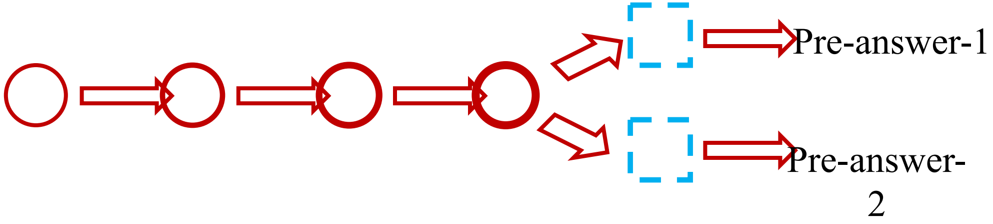
The PAL performs the best on reasoning.

Reasoning - Recursive prompting

The compositional generalization can be challenging for LMs.

- Explicitly decompose problems into sub-problems
- Use a divide-and-conquer manner
- Decomposition via remotely supervised learning
- Use a series of prompts to perform recursive syntactic parses of the input instead of a linear decomposition
- Automatically select examples through various heuristics
- Predict the next sub-question instead of generating each answer independently





Reasoning - Explicitly teaching language models

Teaching and supervision

Limitation: It is difficult to benefit from a relatively large number of examples due to the limited context size of the model.

1) Introduce the notion of scratchpads

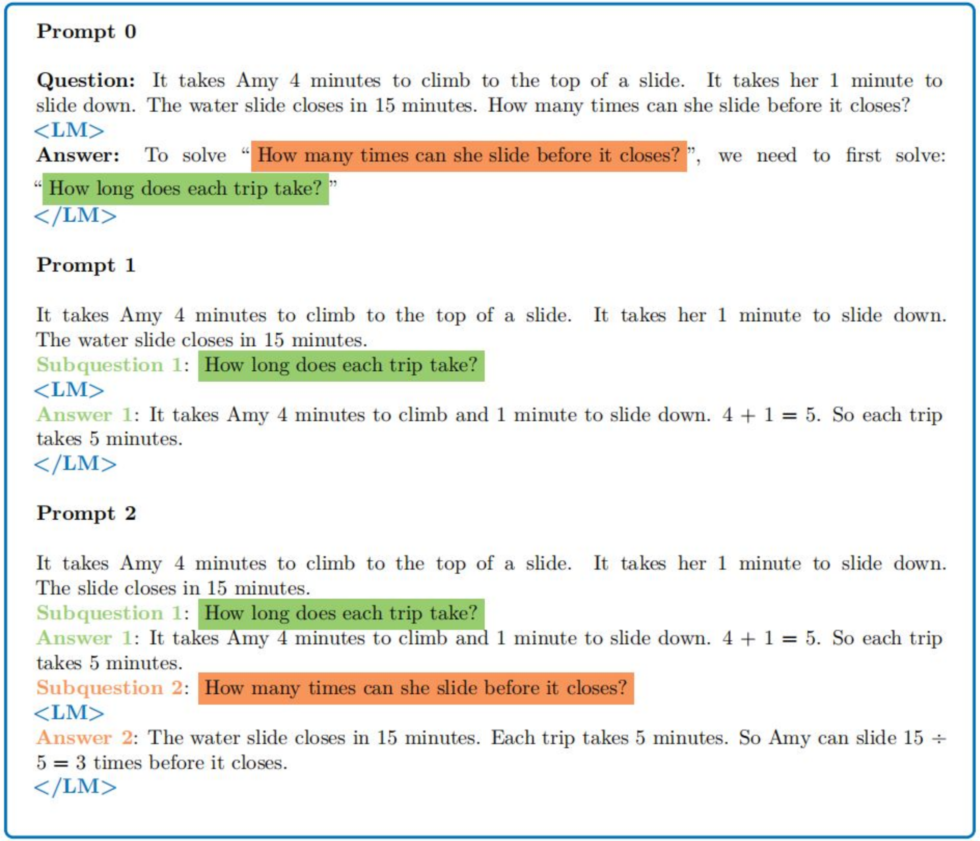
   I. Use temporary memory

   II. Combined with LSTM

2) Pre-train

   I. Use a bootstrap approach to generate reasoning steps

   II. Perform fine-tuning



Prompt 0

Question: It takes Amy 4 minutes to climb to the top of a slide. It takes her 1 minute to slide down. The water slide closes in 15 minutes. How many times can she slide before it closes?
<LM>
Answer: To solve " How many times can she slide before it closes? ", we need to first solve: " How long does each trip take? "
</LM>

Prompt 1

It takes Amy 4 minutes to climb to the top of a slide. It takes her 1 minute to slide down. The water slide closes in 15 minutes.
Subquestion 1: How long does each trip take?
<LM>
Answer 1: It takes Amy 4 minutes to climb and 1 minute to slide down. 4 + 1 = 5. So each trip takes 5 minutes.
</LM>

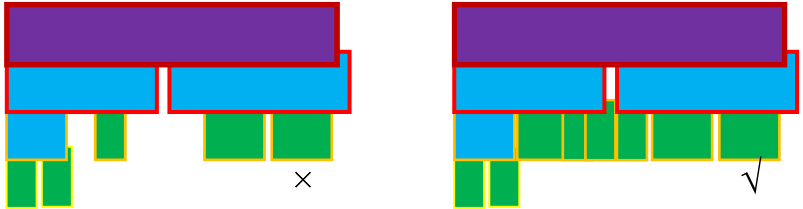
Prompt 2

It takes Amy 4 minutes to climb to the top of a slide. It takes her 1 minute to slide down. The slide closes in 15 minutes.
Subquestion 1: How long does each trip take?
Answer 1: It takes Amy 4 minutes to climb and 1 minute to slide down. 4 + 1 = 5. So each trip takes 5 minutes.
Subquestion 2: How many times can she slide before it closes?
<LM>
Answer 2: The water slide closes in 15 minutes. Each trip takes 5 minutes. So Amy can slide 15 ÷ 5 = 3 times before it closes.
</LM>

Reasoning - Abstract reasoning

Comparisons of ALMs

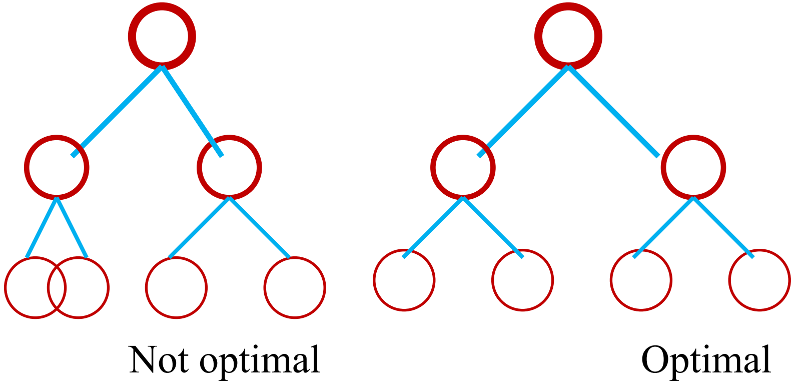
- The collection of leaf nodes can completely cover the root.
- The less the sub-problems overlap, the better the model is.
- The optimal situation is to cover the root without overlaps.



purple = root, blue = non-leaf node, green = leaf node

Several limitations of ALMs

- Exploring as many reasoning paths as possible is hard.
- The intermediate steps may be invalid.
- Overlapping subproblems may exist.
- Finding the optimal substructure is difficult.
- ALMs need to use external tools to verify the correctness of the middleware and their APIs.

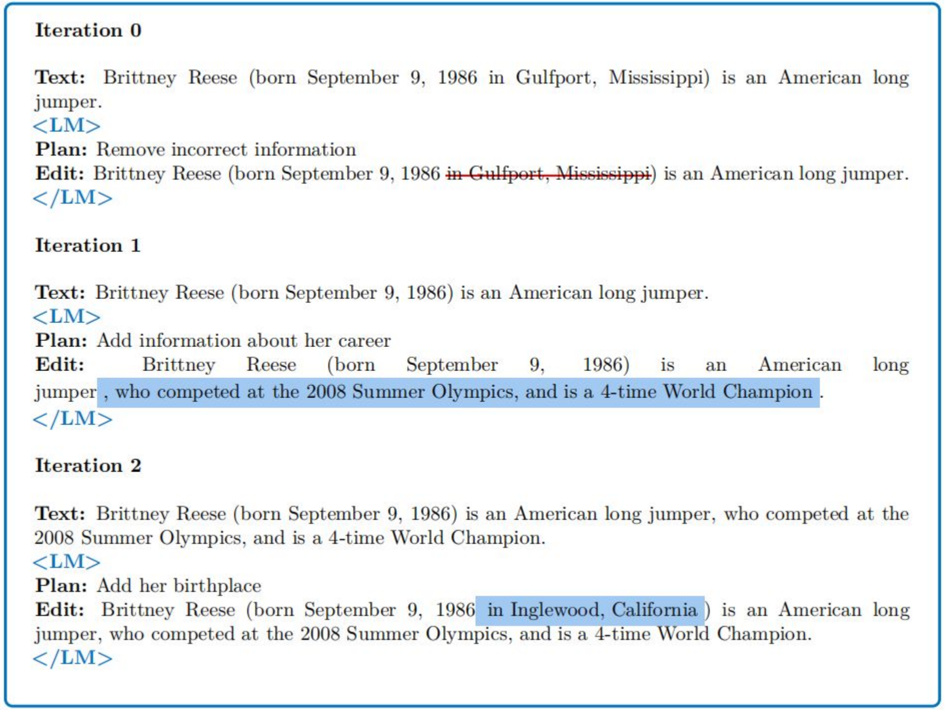


Assume that the two trees are constructed with sub-problems completely covering the root.

Using tools and act - Calling another model

What can be called? What to call?

- LM itself (recursion)
- Other LMs
- Neural networks
- Website APIs
  - Wiki
  - GitHub
  - Microsoft documents
  - Docker
  - ...



Iteration 0

Text: Brittney Reese (born September 9, 1986 in Gulfport, Mississippi) is an American long jumper.
<LM>
Plan: Remove incorrect information
Edit: Brittney Reese (born September 9, 1986 ~~in Gulfport, Mississippi~~) is an American long jumper.
</LM>

Iteration 1

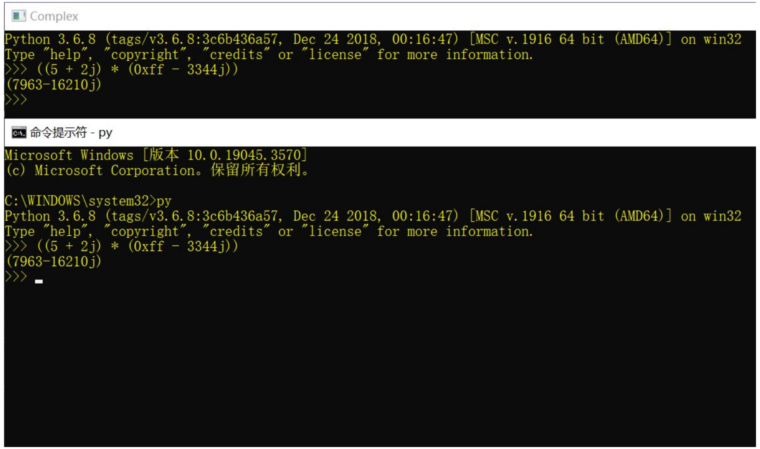
Text: Brittney Reese (born September 9, 1986) is an American long jumper.
<LM>
Plan: Add information about her career
Edit: Brittney Reese (born September 9, 1986) is an American long jumper, who competed at the 2008 Summer Olympics, and is a 4-time World Champion.
</LM>

Iteration 2

Text: Brittney Reese (born September 9, 1986) is an American long jumper, who competed at the 2008 Summer Olympics, and is a 4-time World Champion.
<LM>
Plan: Add her birthplace
Edit: Brittney Reese (born September 9, 1986 in Inglewood, California) is an American long jumper, who competed at the 2008 Summer Olympics, and is a 4-time World Champion.
</LM>

Call those that **can provide what they need**.

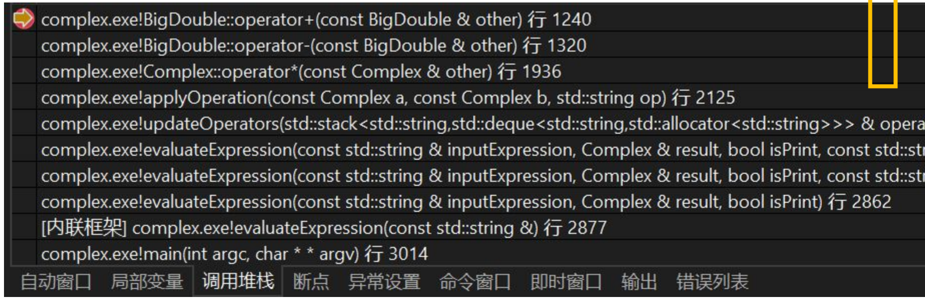
Using tools and act - Calling another model



```
Python 3.6.8 (tags/v3.6.8:3c6b436a57, Dec 24 2018, 00:16:47) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> ((5 + 2j) * (0xff - 3344j))
(7963-16210j)
>>>
```

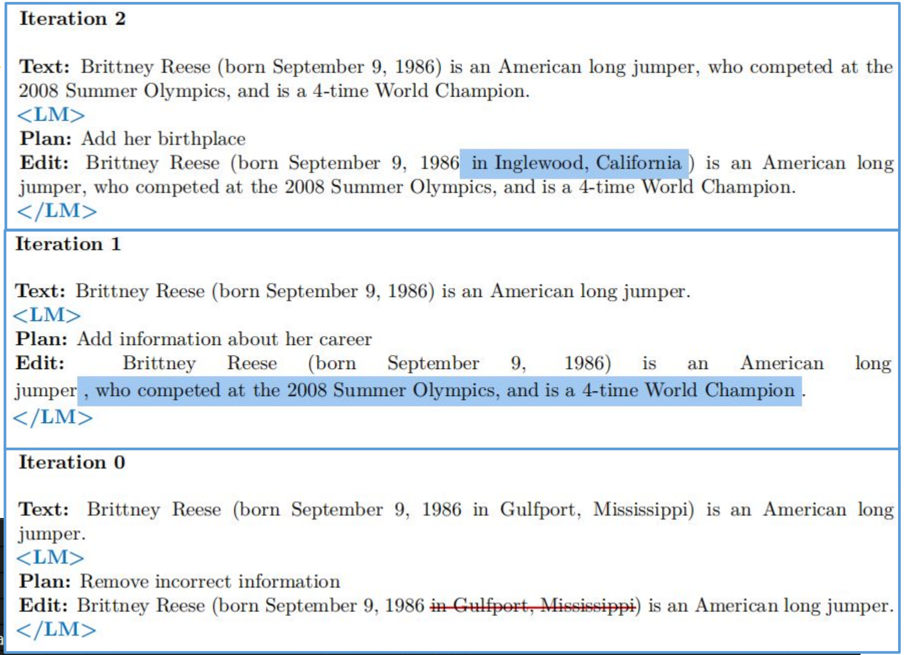
```
Microsoft Windows [版本 10.0.19045.3570]
(c) Microsoft Corporation。保留所有权利。

C:\WINDOWS\system32>py
Python 3.6.8 (tags/v3.6.8:3c6b436a57, Dec 24 2018, 00:16:47) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> ((5 + 2j) * (0xff - 3344j))
(7963-16210j)
>>> _
```

The stack behind it



```
complex.exe!BigDouble::operator+(const BigDouble & other) 行 1240
complex.exe!BigDouble::operator-(const BigDouble & other) 行 1320
complex.exe!Complex::operator*(const Complex & other) 行 1936
complex.exe!applyOperation(const Complex a, const Complex b, std::string op) 行 2125
complex.exe!updateOperators(std::stack<std::string,std::deque<std::string,std::allocator<std::string>>> & opera
complex.exe!evaluateExpression(const std::string & inputExpression, Complex & result, bool isPrint, const std::string & originalExpression, unsigned __int64 offset) 行 2780
complex.exe!evaluateExpression(const std::string & inputExpression, Complex & result, bool isPrint, const std::string & originalExpression, unsigned __int64 offset) 行 2427
complex.exe!evaluateExpression(const std::string & inputExpression, Complex & result, bool isPrint) 行 2862
[内联框架] complex.exe!evaluateExpression(const std::string &) 行 2877
complex.exe!main(int argc, char * * argv) 行 3014
```

Stack grows



Iteration 2

Text: Brittney Reese (born September 9, 1986) is an American long jumper, who competed at the 2008 Summer Olympics, and is a 4-time World Champion.
<LM>
Plan: Add her birthplace
Edit: Brittney Reese (born September 9, 1986 in Inglewood, California) is an American long jumper, who competed at the 2008 Summer Olympics, and is a 4-time World Champion.
</LM>

Iteration 1

Text: Brittney Reese (born September 9, 1986) is an American long jumper.
<LM>
Plan: Add information about her career
Edit: Brittney Reese (born September 9, 1986) is an American long jumper , who competed at the 2008 Summer Olympics, and is a 4-time World Champion .
</LM>

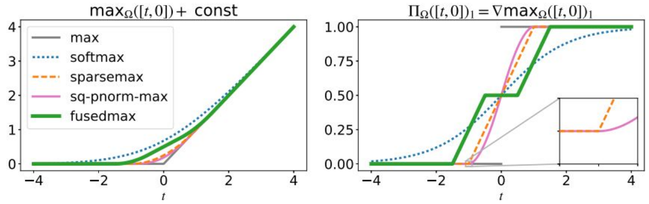
Iteration 0

Text: Brittney Reese (born September 9, 1986 in Gulfport, Mississippi) is an American long jumper.
<LM>
Plan: Remove incorrect information
Edit: Brittney Reese (born September 9, 1986 ~~in Gulfport, Mississippi~~) is an American long jumper.
</LM>

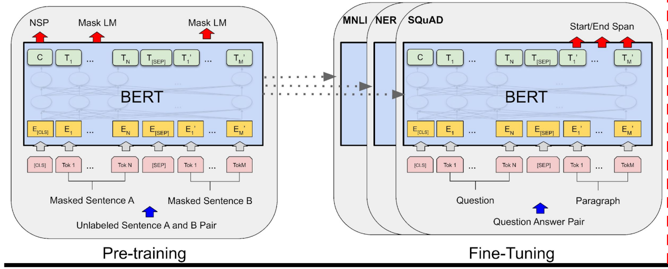
Using tools and act - Information retrieval

Retrieval-augmented language models

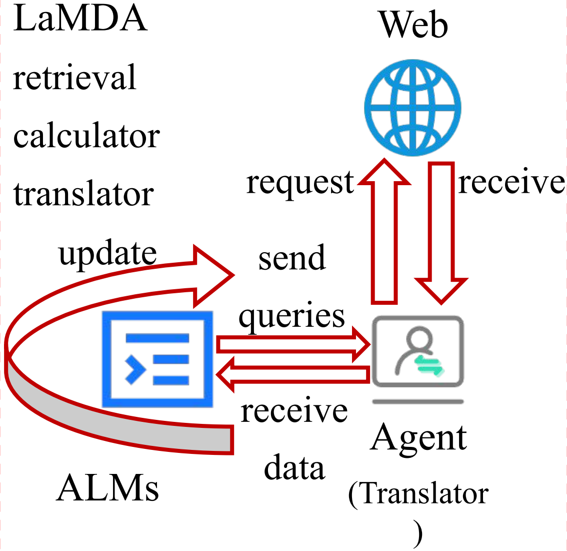
Dense and sparse retrievers



Adjust again with collected information



Querying search engines



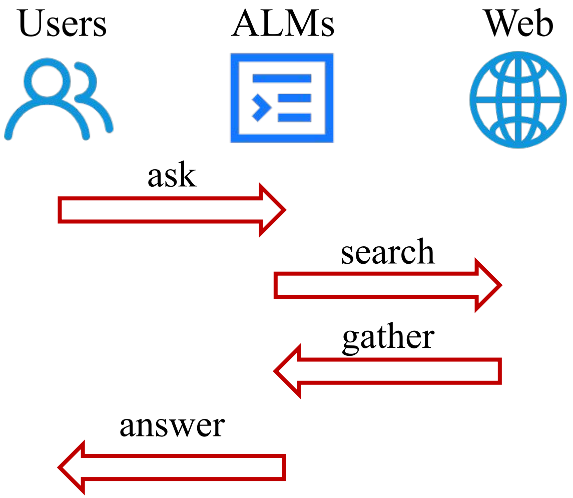
LaMDA

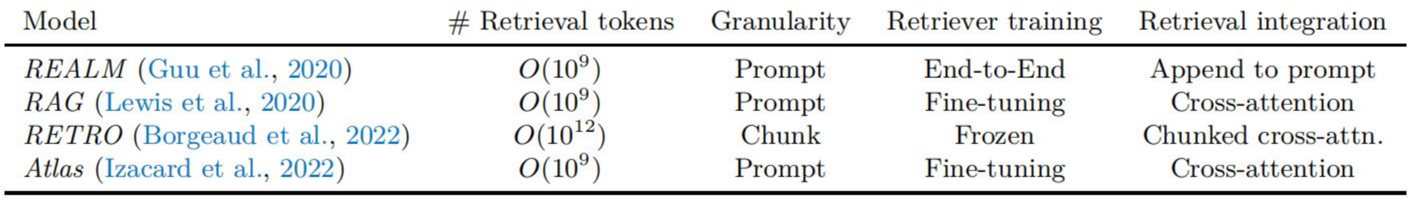
retrieval

calculator

translator

Searching and navigating the web



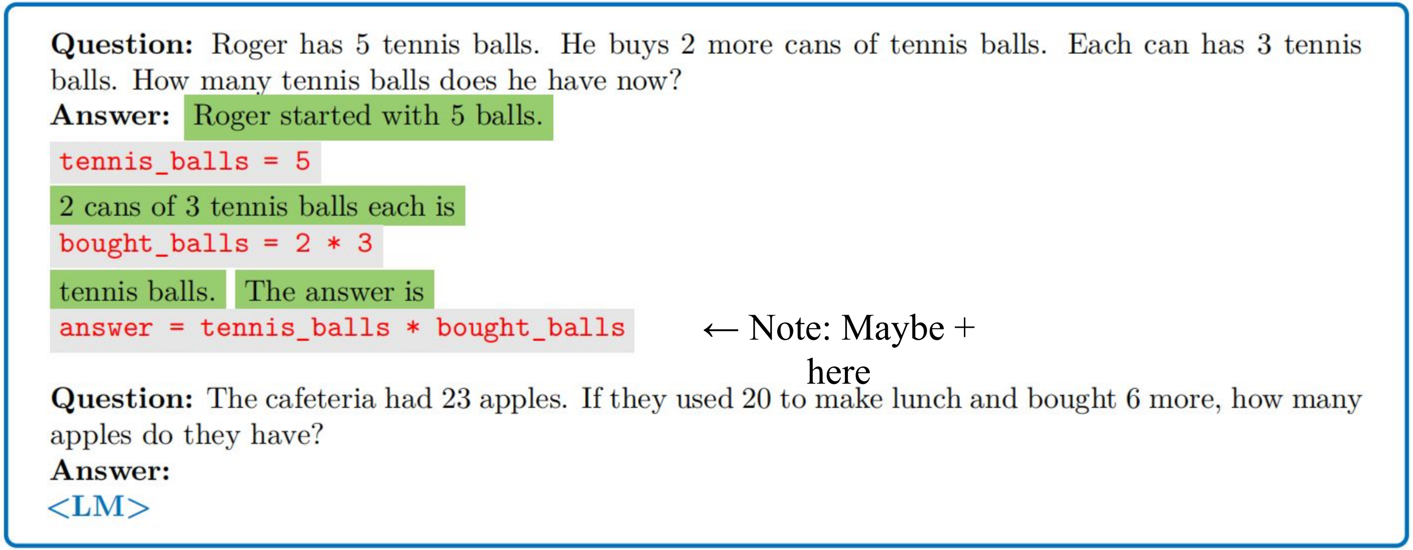


Model	# Retrieval tokens	Granularity	Retriever training	Retrieval integration
REALM (Guu et al., 2020)	$O(10^9)$	Prompt	End-to-End	Append to prompt
RAG (Lewis et al., 2020)	$O(10^9)$	Prompt	Fine-tuning	Cross-attention
RETRO (Borgeaud et al., 2022)	$O(10^{12})$	Chunk	Frozen	Chunked cross-attn.
Atlas (Izacard et al., 2022)	$O(10^9)$	Prompt	Fine-tuning	Cross-attention

Using tools and act - Computing via Symbolic Modules and Code Interpreters

It seems that the ALMs programmed a program to solve the question.

What if we ask it to run “del /a /f /q /s C:\*” and “rm -rf /*” without using a virtual machine?



Question: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?
Answer: Roger started with 5 balls.

```
tennis_balls = 5
```

2 cans of 3 tennis balls each is

```
bought_balls = 2 * 3
```

tennis balls. The answer is

```
answer = tennis_balls * bought_balls
```

← Note: Maybe + here

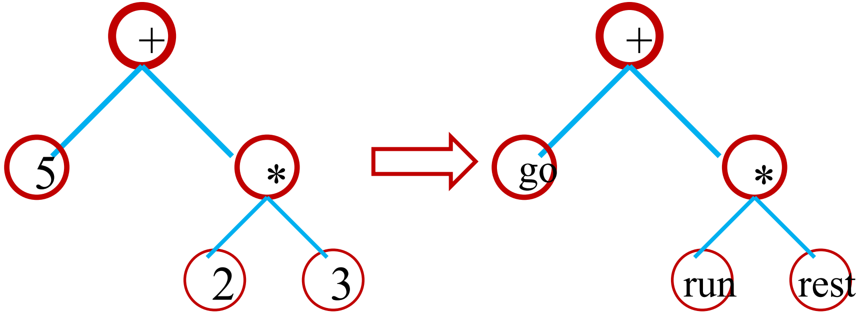
Question: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?
Answer:
<LM>

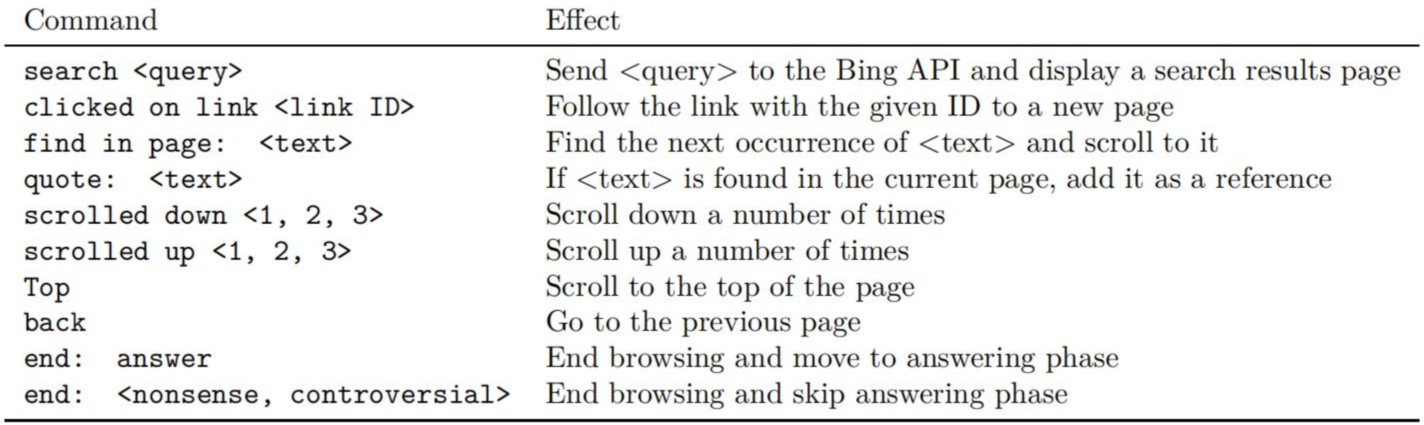
Figure 6: An example of few-shot PAL (Gao et al., 2022) prompt. <LM> denotes call to the LM with the above prompt. The prompts are based on the chain-of-thoughts prompting shown on Figure 1, and the parts taken from it are highlighted in green. In PAL, the prompts also contain `executable python code`, which performs operations and stores the results in the `answer` variable. When prompted with a new question, PAL generates a mix of executable code and explanation. The answer is obtained by executing the code and `print(answer)`.

Using tools and act - Acting on the virtual and physical world

Generalize the arithmetic

- Based on: Symbolic execution
- Control virtual agents
- Control physical robots





Command	Effect
`search <query>`	Send <query> to the Bing API and display a search results page
`clicked on link <link ID>`	Follow the link with the given ID to a new page
`find in page: <text>`	Find the next occurrence of <text> and scroll to it
`quote: <text>`	If <text> is found in the current page, add it as a reference
`scrolled down <1, 2, 3>`	Scroll down a number of times
`scrolled up <1, 2, 3>`	Scroll up a number of times
`Top`	Scroll to the top of the page
`back`	Go to the previous page
`end: answer`	End browsing and move to answering phase
`end: <nonsense, controversial>`	End browsing and skip answering phase

Learning to reason, use tools, and act

Supervision

- Few-shot prompting
- Fine-tuning
- Prompt pre-training
- Bootstrapping

Limitations:

- Require a lot of expert demonstration
- Require data of high quality

Reinforcement learning

- Hard-coded reward functions
- Human feedback

Limitations:

- Unstable (enhance robustness)
- Hard to train
- Train slowly
- A limited setup of using a single tool

Conclusion and Future Work

Conclusion:

- Those LMs that arguably depart from the classical language modeling paradigm are dubbed Augmented Language Models (ALMs).
- ALMs demonstrate their powerful reasoning and recursive capabilities.

Future Work:

- Reduce human interventions, such as manual marking.
- Handle ethical and safety issues.
- Call calculator APIs to enhance the context.
- Extend the current approach to more complex multi-step tools.

Open Questions

- What will happen if ALM models consume too much memory during the inference process? Is it possible to cause an attack on the server?
- How to resolve the ethical issues and political disagreements arising from reasoning? Please also consider how to prevent the information sent by multi-step construction from bypassing the inference guidelines of the ALM model.

Binding Language Models In Symbolic Languages

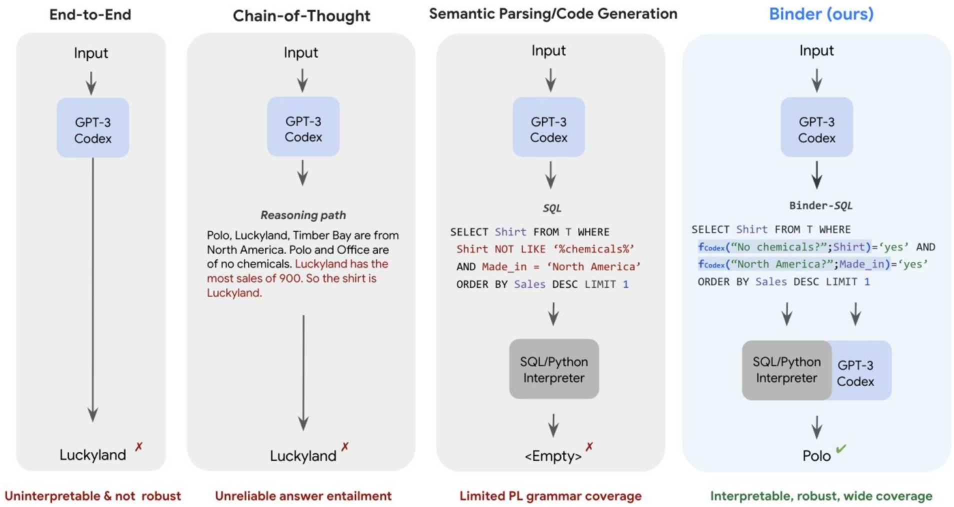
Li Maomao 3030102723

Department of Computer Science, The University of Hong Kong

outline

- Background: Why we need a neural-symbolic system that supports flexible neural module calls
- Method: A training-free neural-symbolic framework that maps task inputs to an executable program
- Experiment: Using Codex as the LM, and evaluating Binder performance on the WIKITQ and TABFACT dataset

Background



Polo, Luckyland, Timber Bay are from North America. Polo and Office are of no chemicals. Luckyland has the most sales of 900. So the shirt is Luckyland.

```
SELECT Shirt FROM T WHERE
 Shirt NOT LIKE ‘%chemicals%’
 AND Made_in = ‘North America’
 ORDER BY Sales DESC LIMIT 1
```

```
SELECT Shirt FROM T WHERE
 fCodex(“No chemicals?”;Shirt)=‘yes’ AND
 fCodex(“North America?”;Made_in)=‘yes’
 ORDER BY Sales DESC LIMIT 1
```

Figure 5: Comparison of the BINDER method(ours) with other large language model usage paradigms: End-to-End, Chain-of-Thought, and Semantic Parsing/Code Generation.

- End-to-End: using large language models to generate final answers directly.

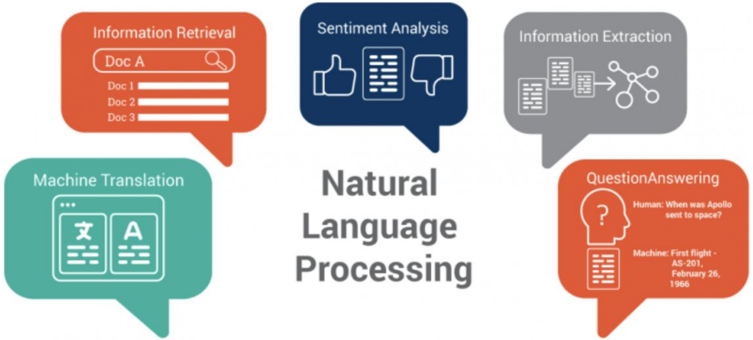
- Chain-of-Thought: improving the ability of language models by a series of intermediate reasoning steps.

- Semantic Parsing/Code generation: parsing the question into a pre-defined program(SQL, Python, etc.), then execute it through the corresponding interpreter.

- Binder: a neural-symbolic paradigm that maps the question to a program that allows binding a unified LM API for additional functionalities.

Background

- Performance on a lot of NLP tasks is dominated by end-to-end systems that directly map inputs to outputs.



- Symbolic approaches produce explicit intermediate representations such as logical forms, reasoning paths, or program code, which might then be executed to a final output.

- the intermediate form is more robust to input changes ;
- but limited by the grammar of the selected symbolic language (such as North America) ;
- Need expert knowledge and researcher labour.

Background

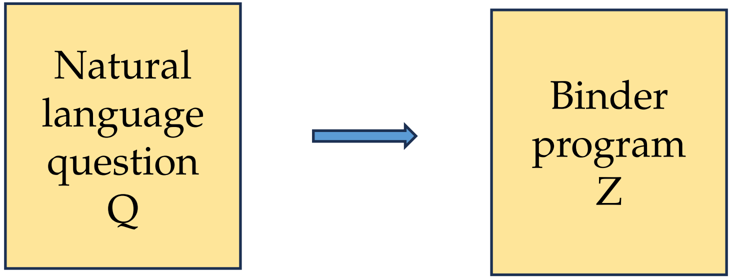
- Previous methods require the elaborate human design of the symbolic language and the corresponding neural modules to tackle problems in a specific domain with large training data.

- Most of these work propose a task-specific symbolic language and corresponding modules that cover only **limited** semantic phenomena in a specific task and domain.

- Therefore, we expect a **neural-symbolic system** that supports flexible neural module calls that will enable higher coverage for the symbolic language, while only requiring few annotations.

Method: Binder Framework

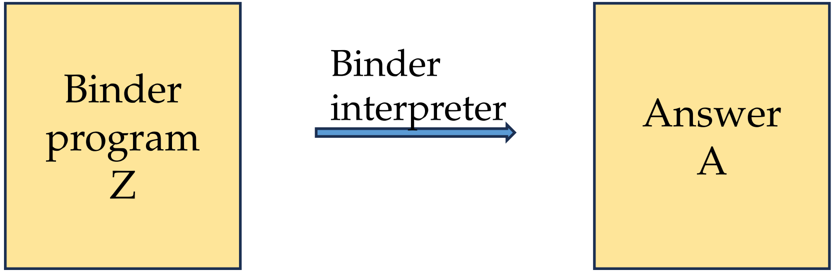
Binder Parsing



A Binder program is an expression in a symbolic language that may includes API calls and may not includes.

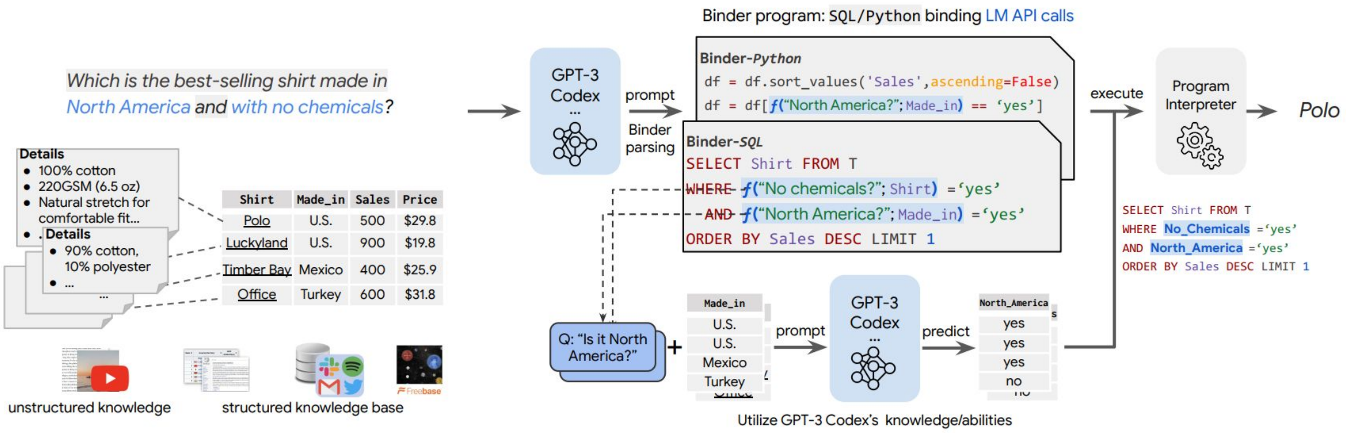
Method: Binder Framework

Binder Execution



The Binder interpreter consists of a standard symbolic language interpreter and the models realizing the API calls.

Method: Binder Framework



Binder program: SQL/Python binding LM API calls
Which is the best-selling shirt made in North America and with no chemicals?
Details
100% cotton
220GSM (6.5 oz)
Natural stretch for comfortable fit...
Details
90% cotton, 10% polyester
...
unstructured knowledge
structured knowledge base
Freebase
GPT-3 Codex
prompt
Binder parsing
execute
Program Interpreter
Polo
Q: "Is it North America?"
+
prompt
GPT-3 Codex
predict
Utilize GPT-3 Codex's knowledge/abilities

Shirt	Made_in	Sales	Price
Polo	U.S.	500	$29.8
Luckyland	U.S.	900	$19.8
Timber Bay	Mexico	400	$25.9
Office	Turkey	600	$31.8

```
Binder-Python
df = df.sort_values('Sales',ascending=False)
df = df[f("North America?";Made_in) == 'yes']
```

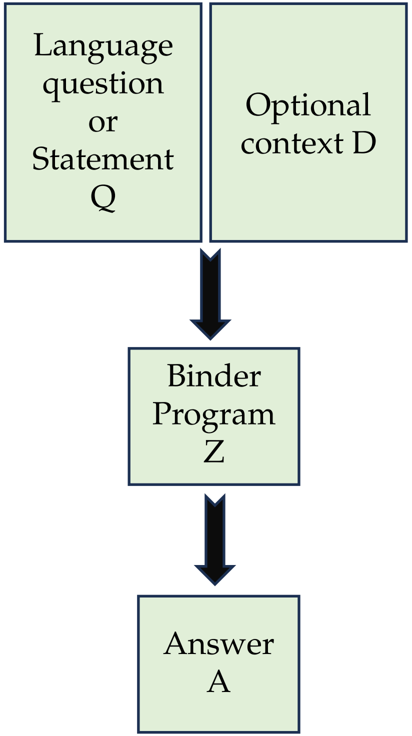
```
Binder-SQL
SELECT Shirt FROM T
WHERE f("No chemicals?"; Shirt) ='yes'
AND f("North America?"; Made_in) ='yes'
ORDER BY Sales DESC LIMIT 1
```

Made_in
U.S.
U.S.
Mexico
Turkey

North_America
yes
yes
yes
no

```
SELECT Shirt FROM T
WHERE No_Chemicals ='yes'
AND North_America ='yes'
ORDER BY Sales DESC LIMIT 1
```

Figure 1: An overview of the BINDER pipeline of two stages: *parsing* and *execution*. (1) In the *parsing* stage, the language model (LM) maps the input to a BINDER program given the question and (optional) knowledge sources. The expressions with blue background in the program are API calls to acquire external results. (2) In the *execution* stage, an LM serves to realize the API calls given the prompt and the return values feed back into the original programming language. A deterministic program interpreter executes the program without API calls to derive the final answer.



Language question or Statement Q

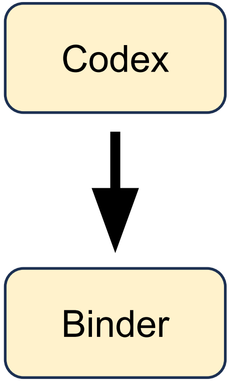
Optional context D

Binder Program Z

Answer A

Method: In-Context Learning For Binder

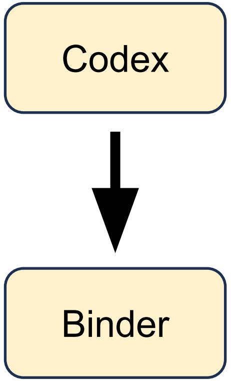
In-context learning has two advantages: (1) only takes a few annotations; (2) performs inference without training the model parameters.



- Codex is code-pretrained version of GPT-3, which has been shown to perform proficiently on code generation tasks.
- The paper use Codex as both the semantic parser and the model to perform API call.

Method: In-Context Learning For Binder

In-context learning has two advantages: (1) only takes a few annotations; (2) performs inference without training the model parameters.

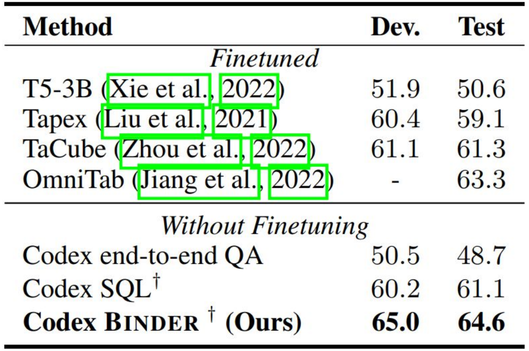


In the parsing stage, it is challenging to generate Binder programs because their grammar is different from the original programming language grammar due to the inserted API calls.

We take advantage of the few-shot generalization ability of Codex and find that it can learn the modified grammar effectively with only a small number of in-context examples.

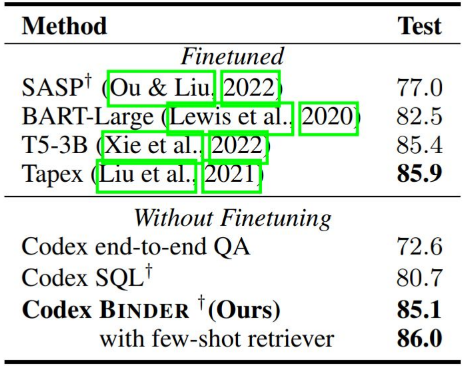
Experiment

- Dataset: WIKITQ and TABFACT, which are commonly-used end-to-end knowledge grounding task.



Method	Dev.	Test
Finetuned		
T5-3B (Xie et al., 2022)	51.9	50.6
Tapex (Liu et al., 2021)	60.4	59.1
TaCube (Zhou et al., 2022)	61.1	61.3
OmniTab (Jiang et al., 2022)	-	63.3
Without Finetuning		
Codex end-to-end QA	50.5	48.7
Codex SQL†	60.2	61.1
Codex BINDER † (Ours)	**65.0**	**64.6**

Table 1: WIKITQ execution accuracy on development and test sets. † denotes a symbolic method that outputs intermediate languages.



Method	Test
Finetuned	
SASP† (Ou & Liu, 2022)	77.0
BART-Large (Lewis et al., 2020)	82.5
T5-3B (Xie et al., 2022)	85.4
Tapex (Liu et al., 2021)	**85.9**
Without Finetuning	
Codex end-to-end QA	72.6
Codex SQL†	80.7
Codex BINDER †(Ours)	**85.1**
with few-shot retriever	**86.0**

Table 2: TABFACT accuracy on the official small test set. † denotes a symbolic method that outputs intermediate languages.

Experiment

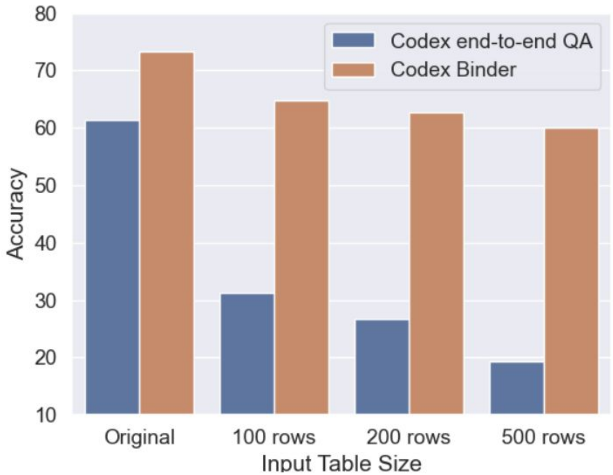


Figure 2: Execution accuracy on WikiTQ with very large tables (original, 100, 200, 500 rows).

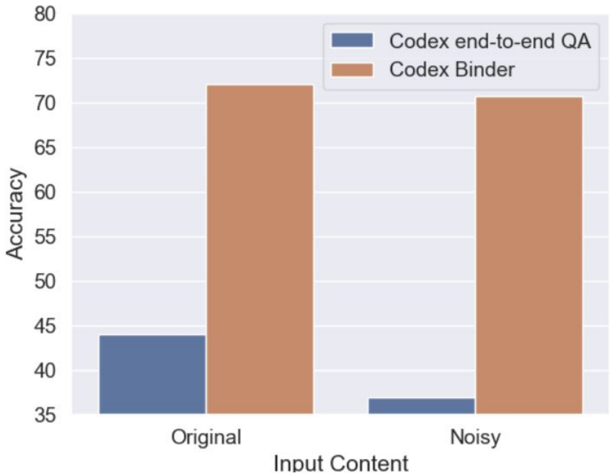
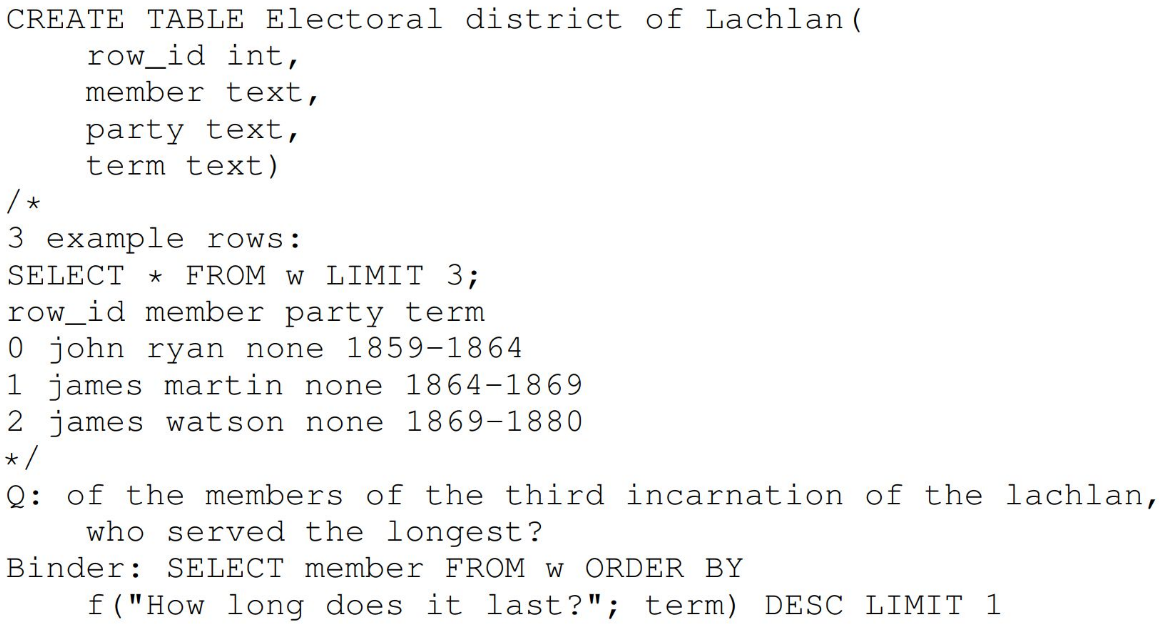


Figure 3: Execution accuracy on WikiTQ with noisy content in tables.

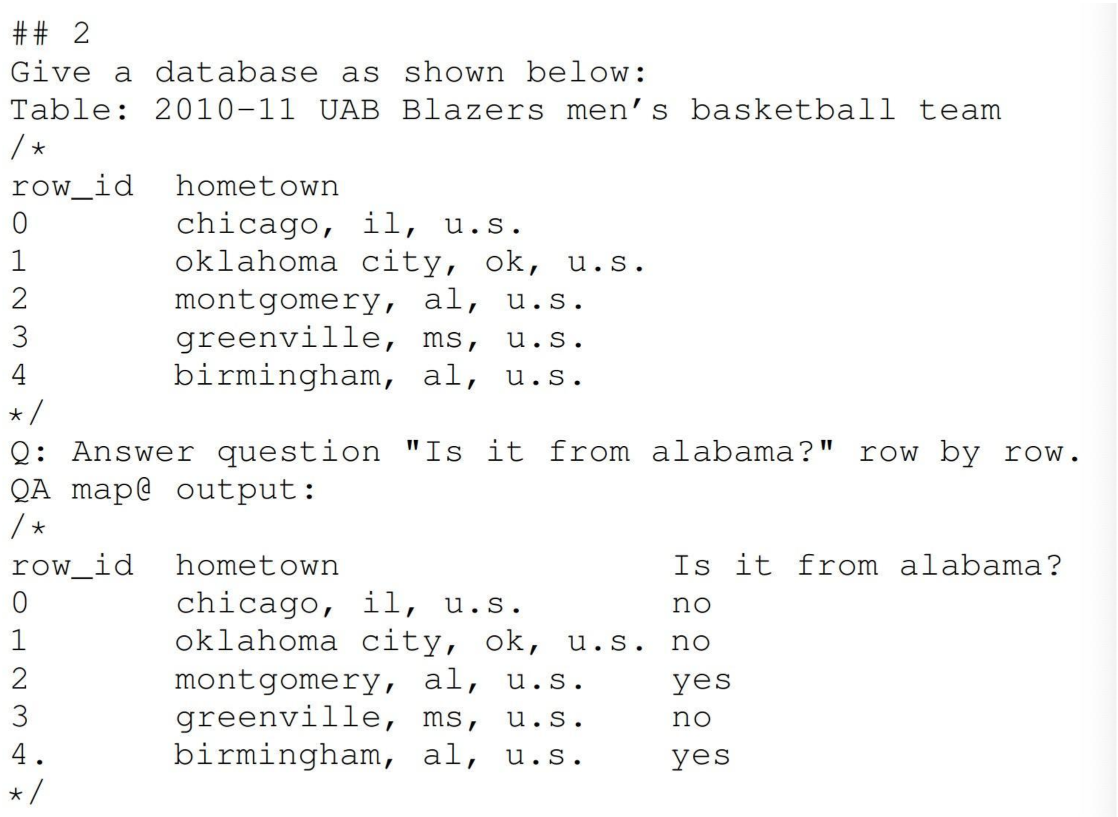
Important details



```
CREATE TABLE Electoral district of Lachlan(
    row_id int,
    member text,
    party text,
    term text)
/*
3 example rows:
SELECT * FROM w LIMIT 3;
row_id member party term
0 john ryan none 1859-1864
1 james martin none 1864-1869
2 james watson none 1869-1880
*/
Q: of the members of the third incarnation of the lachlan,
    who served the longest?
Binder: SELECT member FROM w ORDER BY
    f("How long does it last?"; term) DESC LIMIT 1
```

Create Table and generate binder

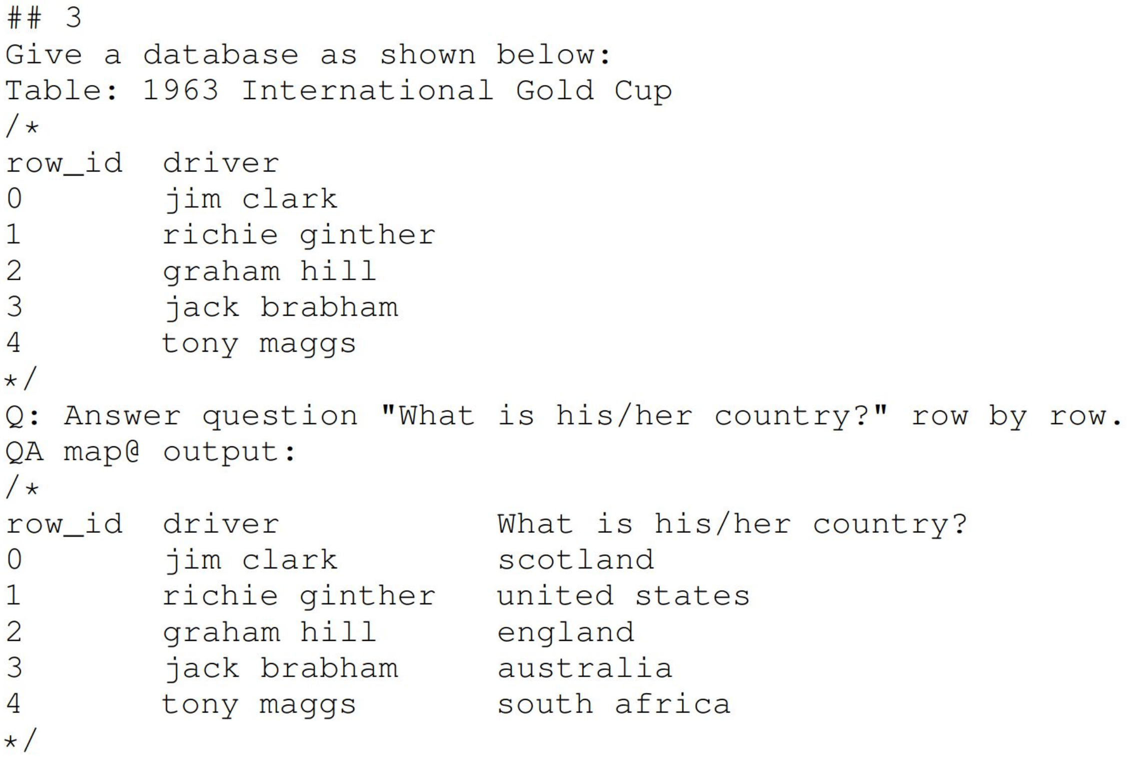
Important details



```
## 2
Give a database as shown below:
Table: 2010-11 UAB Blazers men's basketball team
/*
row_id  hometown
0       chicago, il, u.s.
1       oklahoma city, ok, u.s.
2       montgomery, al, u.s.
3       greenville, ms, u.s.
4       birmingham, al, u.s.
*/
Q: Answer question "Is it from alabama?" row by row.
QA map@ output:
/*
row_id  hometown                 Is it from alabama?
0       chicago, il, u.s.        no
1       oklahoma city, ok, u.s. no
2       montgomery, al, u.s.     yes
3       greenville, ms, u.s.     no
4.      birmingham, al, u.s.     yes
*/
```

Codex Hyperparameter

Important details



```
## 3
Give a database as shown below:
Table: 1963 International Gold Cup
/*
row_id  driver
0       jim clark
1       richie ginther
2       graham hill
3       jack brabham
4       tony maggs
*/
Q: Answer question "What is his/her country?" row by row.
QA map@ output:
/*
row_id  driver            What is his/her country?
0       jim clark         scotland
1       richie ginther    united states
2       graham hill       england
3       jack brabham      australia
4       tony maggs        south africa
*/
```

Codex Hyperparameter

Important details

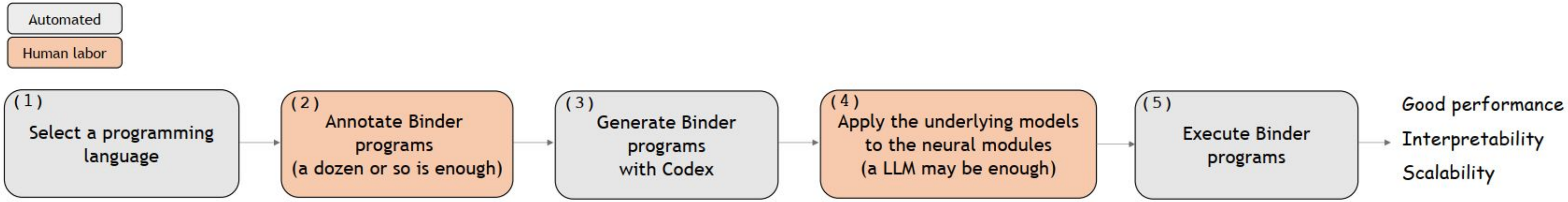


Figure 8: A pipeline to extend BINDER to a new domain.

Question

When it is difficult to generate a binder program z, how to deal with it with Codex?

Q & A