A Additional Related Works

Methods	Tool Numbers	Tool Categories	# Tool/Task	Reasoning	Instruction Type	Task
Single-Tool Methods						
CoT [65]	1	-	1	Generation	Prompting	QA
Lila [38]	1	math/code	1	Generation	Prompting	MathQA
Program-of-Thought [6]	1	code	1	Generation	Prompting	TabQA
Code4Struct [63]	1	code	1	Generation	Prompting	Event Extraction
PAL [13]	1	code	1	Generation	Prompting	MathQA
MathPrompt [15]	1	code	1	Generation	Prompting	MathQA
ToolFormer [56]	5	Basic	1	Generation	PR & FT	QA
GraphToolFormer [77]	5	Graph	1	Human Info	PR & FT	Graph
Talm [46]	-	Basic	1	Generation	PR & FT	QA
Multi-Tool Methods						
WebGPT [40]	10	Web Operation	>1	Feedback	Fine-tuning	QA
HuggingGPT [57]	>10	Vision	>1	Human Info	Prompting	VQA
Chameleon [32]	>10	code, nlp, cv	>1	Human Info	Prompting	ScienceQA, TabQA
GeneGPT [20]	38	NCBI APIs	>1	Generation	Prompting	Gene Tasks
ART [45]	8	code/math/retriever	>1	Human Feedback	Prompting	BigBench
ReAct [76]	3	retriever	>1	Feedback	PR & FT	QA, AlfWorld, WebShop
MM-ReAct [73]	>10	vision	>1	Feedback	Prompting	CV tasks
Visual ChatGPT [69]	>10	vision	>1	Feedback	Prompting	CV tasks

Table 6: A comparison of methods that leverage LLMs for Tool-use.

We list the state-of-the-art related works in tool-augmented LLMs in Table 6. All of them can be categorized into two groups: (1) single-tool methods, that focus on making a single API call perfect in the solution; (2) multi-tool methods, that emphasize more on studying how to compose different tools together to solve a challenging problem. ToolQA is more suitable for the evaluation of the second category to test the inherent logical reasoning behind different tools. Additionally, there exist other notable contributions [64, 25, 59] within the realm of decision-making that specifically emphasize the planning capabilities of expansive language models. These endeavors can be regarded as methods affiliated with tools, wherein the actions within generated plans are analogous to distinct tools utilized for specific purposes.

B Data Sources

B.1 Different Data Source Introduction

- **Flight Status (2022-2023)**[5] contains almost all flight information of airlines between 2022 and 2023, which is too contemporary for LLMs' internal knowledge.
- **Daily Coffee Price (2000-2022)**[6] contains the daily price of coffee, ranging from 2000 to 2022, where the information is too contemporary and detailed for LLMs' internal knowledge.
- **Yelp Business Data**[7] is a subset of Yelp's business data across 8 metropolitan areas in the USA and Canada, where the information is too detailed for LLMs' internal knowledge.
- **Airbnb Open Data**[8] is a subset of Airbnb activities in New York, where the information is too detailed for LLMs' internal knowledge.
- **DBLP Citation Network (V14)**[9] constructs the graph based on the records after 2020. The author-author and paper-paper relations are formulated as two separate graphs.
- **GSM8k**[10] is a dataset of 8.5K high-quality linguistically diverse grade school math word problems. We sample the questions from the error cases made by ChatGPT on the original dataset to make sure that the questions cannot be easily handled with its internal knowledge.
- **SciREX**[11] is a challenging dataset for document-level information extraction based on a collection of full-length machine-learning scientific papers.

[5] https://www.kaggle.com/datasets/robikscube/flight-delay-dataset-20182022?select=Combined_Flights_2022.csv
[6] https://www.kaggle.com/datasets/psycon/daily-coffee-price
[7] https://www.kaggle.com/datasets/yelp-dataset/yelp-dataset?select=yelp_academic_dataset_business.json
[8] https://www.kaggle.com/datasets/arianazmoudeh/airbnbopendata
[9] https://www.aminer.org/citation
[10] https://github.com/openai/grade-school-math
[11] https://github.com/allenai/SciREX

- **Agenda** is our own synthetic dataset to model the real-world personal agenda data. To avoid the privacy issue, we first create names, events, and dates with ChatGPT and then randomly compose them to form 10000 different records. To create a pure-text personal agenda corpus, we feed each of the records into ChatGPT, containing generated agenda for virtual characters. More Details can be seen in Appendix B.2.

B.2 Generation Details of Agenda Dataset

As mentioned in § 3.2, personal or private data serves as a significant external knowledge source. There exist applications that have been designed with plugins and external tools specifically querying this type of data, such as AI personal assistants on daily agenda. Nevertheless, we recognize that this data often intersects with sensitive areas, and hence, privacy concerns are paramount. To address these issues, we automatically synthesize a personal agenda corpus. This not only ensures that the large language models (LLMs) have not been previously exposed to the data but also eliminates any possibility of them inadvertently memorizing the information within their internal knowledge.

In the synthetically generated personal agenda corpus, each entry follows the pattern: "`NAME` performs `EVENT` at `TIME` on `DATE`", incorporating key elements such as names, events, dates, and time slots. To begin, we employ ChatGPT to virtually generate these elements. More precisely, we create 100 unique names, 10000 distinctive events each associated with corresponding time slots within a day, and span all possible dates from 01/01/2022 through 12/31/2022. Following this, we commence the random assembly of these generated elements to formulate personal agenda entries. For every event-time pair generated, we randomly select from the pool of 100 names and possible dates to construct each record. This process yields a total of 9,494 unique personal agenda entries. To transform this corpus into an accessible external database for model querying, we transcribe each record into a comprehensible natural language description. Prompts designed for agenda data generation are listed in Appendix F.2.

C Easy Question Templates

C.1 Flights

We design the following 10 templates:

- What was the departure time of the `{CARRIER}{NUMBER}` flight from `{ORIGIN}` to `{DEST}` on `{ORIGIN}`?
- Was the flight `{CARRIER}{NUMBER}` from `{ORIGIN}` to `{DEST}` cancelled on `{ORIGIN}`?
- What is the flight number of the `{AIRLINE}` flight from `{ORIGIN}` to `{DEST}` on `{ORIGIN}`?
- How long was the different between the CRS-recorded departure time and actual departure time of the `{CARRIER}{NUMBER}` flight from `{ORIGIN}` to `{DEST}` on `{ORIGIN}`?
- How long did `{CARRIER}{NUMBER}` delay when arrival on `{DEST}`?
- How many extra minutes did the `{CARRIER}{NUMBER}` flight take from `{ORIGIN}` to `{DEST}` on `{ORIGIN}`?
- What was the local arrival time of the `{CARRIER}{NUMBER}` flight from `{ORIGIN}` to `{DEST}` on `{ORIGIN}`?
- What was the CRS-recorded arrival time of the `{CARRIER}{NUMBER}` flight from `{ORIGIN}` to `{DEST}` on `{ORIGIN}`?
- How long was the flight `{CARRIER}{NUMBER}` from `{ORIGIN}` to `{DEST}` on `{ORIGIN}`?
- How many minutes did the `{CARRIER}{NUMBER}` flight take to taxi in on `{DATE}`?

C.2 Coffee

We design the following 8 templates:

- What was the daily coffee price opening on `{DATE}`?
- What was the lowest coffee price on `{DATE}`?
- What was the highest coffee price on `{DATE}`?
- What was the daily coffee price closing on `{DATE}`?
- What was the trading volume of coffee on `{DATE}`?

- What was the percentage change in coffee price on `{DATE}`, based on the difference between the opening and closing prices?
- Was `{DATE}` a bearish or bullish day for coffee price?
- What was the range of coffee price on `{DATE}`, based on the difference between the high and low prices?

C.3 Yelp

We design the following 11 templates for the Yelp dataset:

- What is the address of `{NAME}` in the area of postal code `{POSTAL-CODE}`?
- What city is `{NAME}` located in `{STATE}`?
- What state is `{NAME}` located in?
- What is the postal code of `{NAME}` in the area with postal code `{POSTAL-CODE}`, `{CITY}`, `{STATE}`?
- What is the star rating of `{NAME}` in the area with postal code `{POSTAL-CODE}`, `{CITY}`, `{STATE}`?
- How many reviews does `{NAME}` receive in the area with postal code `{POSTAL-CODE}`, `{CITY}`, `{STATE}`, received?
- Is `{NAME}` still open in the area with postal code `{POSTAL-CODE}`, `{CITY}`, `{STATE}`?
- Does `{NAME}` require appointment in the area with postal code `{POSTAL-CODE}`, `{CITY}`, `{STATE}`?
- What are the hours of operation for `{NAME}` in the area with postal code `{POSTAL-CODE}`, `{CITY}`, `{STATE}`?
- What categories does `{NAME}` belong to, in the area with postal code `{POSTAL-CODE}`, `{CITY}`, `{STATE}`?
- What are the coordinates of `{NAME}` in the area with postal code `{POSTAL-CODE}`, `{CITY}`, `{STATE}`?

C.4 Airbnb

We design the following 10 templates for easy questions on Airbnb dataset:

- What is the host's name for `{NAME}` in `{NEIGHBOURHOOD}`?
- How many days are `{NAME}` (id: `{ID}`) available during a year (365 days)?
- What is the room type of `{NAME}` (id: `{ID}`) in `{NEIGHBOURHOOD}`?
- What is the price of `{NAME}` (id: `{ID}`) in `{NEIGHBOURHOOD}`?
- What is the minimum number of nights for `{NAME}` (id: `{ID}`) in `{NEIGHBOURHOOD}`?
- When did `{NAME}` (id: `{ID}`) in `{NEIGHBOURHOOD}` constructed?
- How many reviews does `{NAME}` (id: `{ID}`) in `{NEIGHBOURHOOD}` have?
- What is the last review date for `{NAME}` (id: `{ID}`) in `{NEIGHBOURHOOD}`?
- What is the review rate number for `{NAME}` (id: `{ID}`) in `{NEIGHBOURHOOD}`?
- What is the average number of reviews per month for `{NAME}` (id: `{ID}`) in `{NEIGHBOURHOOD}`?

C.5 SciREX

We design the following 1 templates for easy questions on SciREX dataset:

- What is the corresponding `{METRIC}` score of the `{METHOD}` method on `{DATASET}` dataset for `{TASK}` task?

C.6 Agenda

We design the following 5 templates for easy questions on Agenda dataset:

- What did `{NAME}` do from `{START-TIME}` to `{END-TIME}` on `{DATE}`?
- Where did `{EVENT}` that `{NAME}` attended take place on `{DATE}`?
- When did `{NAME}` attend `{EVENT}` on `{DATE}`?
- How long did `{NAME}` attend `{EVENT}` on `{DATE}`?
- Who attended `{EVENT}` between `{START-TIME}` and `{END-TIME}` on `{DATE}` in `{LOCATION}`?

C.7 DBLP

We design the following 10 templates for easy questions on DBLP dataset:

- Who are the authors of {TITLE}?
- What organization is {AUTHOR} from?
- How many pages is {TITLE}?
- How many papers did {TITLE} cite in the DBLP citation network?
- How many papers did papers in the DBLP citation network cite {TITLE}?
- How many collaborators does {AUTHOR} have in the DBLP citation network?
- How many papers did {AUTHOR} and {AUTHOR} write together in the DBLP citation network?
- What papers did {AUTHOR} write in the DBLP citation network?
- How many papers did {AUTHOR} write in the DBLP citation network?
- What venue did {AUTHOR} and {AUTHOR} collaborate most in the DBLP citation network?

C.8 GSM8K

The questions are randomly sampled from the ChatGPT errors in GSM8K dataset without following some templates. Thus, we cannot offer any question templates for GSM8K.

D Hard Question Templates

D.1 Flights

- What percentage of the flights from {ORIGIN} were delayed on {FLIGHTDATE}?
- What is the average delay time of all the flights that departed from {ORIGIN} on {FLIGHTDATE}?
- How many flights were diverted on {FLIGHTDATE}?
- How many flights with a distance greater than 500 miles on {FLIGHTDATE}?
- What is the average airtime of the flights from {ORIGIN} to {DEST} host by {AIRLINE}?
- How many flights from {ORIGIN} to {DEST} host by {AIRLINE}?
- What is the average flight time of {CARRIER}{NUMBER}?
- What is the fastest flight from {ORIGIN} to {DEST} on {FLIGHTDATE}?
- What is the average speed of {CARRIER}{NUMBER} from {ORIGIN} to {DEST}?
- What is the total number of flights operated by {AIRLINE} on {FLIGHTDATE}?

D.2 Coffee

- What was the highest coffee price from {START-DATE} to {END-DATE}?
- What was the lowest coffee price from {START-DATE} to {END-DATE}?
- What was the average coffee price from {START-DATE} to {END-DATE}?
- How much did the coffee price change from {START-DATE} to {END-DATE}?
- What was the percentage change in coffee price on {DATE} compared to the previous day?
- On which date from {START-DATE} to {END-DATE} was the difference between the highest and lowest coffee prices the greatest?
- What was the average daily volume of coffee traded from {START-DATE} to {END-DATE}?
- On which date from {START-DATE} to {END-DATE} did the coffee price have the highest increase compared to the previous day?
- How many times from {START-DATE} to {END-DATE} did the coffee price increase compared to the previous day?
- What was the percentage increase in coffee price from {START-DATE} to {END-DATE}?
- What was the coffee price range from {START-DATE} to {END-DATE}?

D.3 Yelp

We design the following 10 templates for hard questions in Yelp Dataset.

- How many {CATEGORY} businesses are there in {CITY}, {STATE}?
- How many businesses are there in {POSTALCODE} area of {CITY}, {STATE}?
- Which {CATEGORY} business has the highest star rating in {CITY}, {STATE}?
- Which {CATEGORY} business has the highest review count in {CITY}, {STATE}?"
- What is the average review counts of businesses within a 5-mile radius from {NAME}?
- Which is the nearest {CATEGORY} business to {NAME}?
- Can you recommend a {CATEGORY} business with the highest star rating within a 5-mile radius of {ADDRESS}?
- How many businesses are not open currently in {CITY}?
- What is the average star rating of {CATEGORY} businesses in {CITY}?
- Which region has most bussinesses in {CITY}, {STATE}?

D.4 Airbnb

We design the following 10 templates for hard questions on Airbnb dataset.

- What is the total price at least if you want to stay at {NAME} in {NEIGHBOURHOOD} for {NUMBER} nights?
- How many airbnbs are there in {NEIGHBOURHOOD}?
- What is the average price of airbnbs in {NEIGHBOURHOOD}?
- What is the average review rates within 5 miles from {NAME} in {NEIGHBOURHOOD}?
- How much proporion of airbnbs in {NEIGHBOURHOOD} have a flexible cancellation policy?
- How much does it cost per night to stay at the most expensive entire home/apt in {NEIGHBOURHOOD}?
- How many airbnbs are there in {NEIGHBOURHOOD} that have a review rate higher than 4?
- Can you recommend me a hotel room with the lowest price in {NEIGHBOURHOOD}?
- Can you recommend me a private room with the highest review rate that can host at least 2 people in {NEIGHBOURHOOD}?
- Can you recommend a shared room with the lowest price within 10 miles from {LONGITUDE} longitude and {LATITUDE} latitude?

D.5 SciREX

We design the following 4 templates for hard questions on SciREX dataset:

- What is the corresponding {METRIC} score of the {METHOD} method on {DATASET} dataset for {TASK} task?
- On which dataset does the {METHOD} method achieve the highest {METRIC} score for {TASK} task?
- Which method achieves the highest {METRIC} score on {DATASET} dataset for {TASK} task?
- On what metrics is the {METHOD} method evaluated on {DATASET} dataset for {TASK} task?
- Which datasets is {METHOD} method evaluated on for {TASK} task?

D.6 Agenda

We design the following 5 templates for hard questions on Agenda dataset:

- How many events happen on {DATE} in the agenda table?
- Who is unavailable between {START-TIME} and {END-TIME} on {DATE} in the agenda table?
- When should I schedule a meeting with {NAME} from 9:00 AM to 6:00 PM on {DATE} in the agenda table?
- What events does {NAME} have on {DATE} in the agenda table?
- How many dates in the agenda table have {NAME} scheduled?

D.7 DBLP

We design the following 10 templates for hard questions on DBLP dataset:

- What keywords does {AUTHOR} focus on most in the DBLP citation network?
- How many people does {AUTHOR-1} need to know at least to know {AUTHOR-2} in the DBLP citation network?
- How many common collaborators does {AUTHOR-1} have with {AUTHOR-2}?
- Which is the most cited paper written by {AUTHOR} in the DBLP citation network?
- Which collaborator does {AUTHOR} have the most citations with in the DBLP citation network?
- Which venue does {AUTHOR} publish the most papers in the DBLP citation network?
- How many accumulated citations do papers collaborated by {AUTHOR-1} and {AUTHOR-2} have in the DBLP citation network?
- How many papers in all do {AUTHOR} and his/her collaborators have in the DBLP citation network?
- Who collaborated with {AUTHOR} most in the DBLP citation network?
- What institutions participated in the study of {TITLE} in the DBLP citation network?

E Code Examples of Programmatic Answer Generation

Below is an example of programmatic answer generation. The example code is answering the question of "What percentage of the flights from {ORIGIN} were delayed on {FLIGHTDATE}?". More details of the programmatic answers can be seen in the public code.

```
def solution(data, flightdate, origin):
    num_total =len(data.loc[(data["FlightDate"] ==flightdate) & (data["Origin"] ==
                                           origin)])
    num_cancelled =len(data.loc[(new_data["FlightDate"] ==flightdate) &
                                           (data["Origin"] ==origin) &
                                           (data["Cancelled"] ==True)])
    if num_cancelled >0:
        question ="What percentage of the flights from {} were delayed on
                                              {}?".format(origin, flightdate)
        answer ="{:.1f}".format(num_cancelled /num_total *100)+"%"
```

F Additional Implementation Details

F.1 Implementation Details

All experiments are conducted on *CPU*: Intel(R) Core(TM) i7-5930K CPU @ 3.50GHz and *GPU*: NVIDIA GeForce RTX A5000 GPUs using python 3.8, Huggingface 4.6.0 and Pytorch 1.10. We keep the parameter top_p $= 1.0$ and temperature $t = 1.0$ for calling ChatGPT APIs [43] for the question generation part.

F.2 Prompts

F.2.1 Prompts for Agenda Data Generation

The prompts used for virtual name generation:

```
<Agenda_Name_Gen> Prompt
You are an AI assistant to answer questions.
Can you list 100 English Names?
```

The prompts used for virtual events generation:

<Agenda_Events _Gen> Prompt

```
You are an AI assistant for text generation.
Generate 100 detailed agenda events, including the event, start time, end time, and
location. Please make the events as diverse as possible and make sure these events
can happen in real life. Make sure the location is a detailed name that may exist
in real life. Make sure the dates are selected from 2022/01/01 to 2023/01/01.

Example:
Doctor's appointment - 9:00 AM - 11:00 AM - ABC Medical Center
Yoga class - 10:30 AM - 11:30 AM - Yoga Studio Downtown

Generate 100 more detailed agendas that do not conflict with the previous ones.
```

The prompts used to convert the agenda records into natural language descriptions:

<Agenda_Gen> Prompt

```
Please use natural language to describe the event in the agenda with the following
information:

Name: NAME
Date: DATE
Event: EVENT
Start Time: START-TIME
End Time: END-TIME
Location: LOCATION
```

F.2.2 Prompts for Methods

The prompts used in ReAct [76]:

<ReAct> Prompt

```
Question: How many extra minutes did the DL1575 flight take from ATL to MCO
    on 2022-01-12?
Thought 1: This is a question related to flights. We need to load the flights
    database.
Action 1: LoadDB[flights]
Observation 1: We have successfully loaded the flights database, including the
    following columns: FlightDate, Airline, Origin, Dest, Cancelled, Diverted,
    CRSDepTime, DepTime, DepDelayMinutes, DepDelay, ArrTime, ArrDelayMinutes,
    AirTime, CRSElapsedTime, ActualElapsedTime, Distance, Year, Quarter, Month,
    DayOfWeek, Marketing_Airline_Network, Operated_or_Branded_Code_Share_Partners,
    DayofMonth, DOT_ID_Marketing_Airline, IATA_Code_Marketing_Airline,
    Flight_Number_Marketing_Airline, Operating_Airline, DOT_ID_Operating_Airline,
    IATA_Code_Operating_Airline, Tail_Number, Flight_Number_Operating_Airline,
    OriginAirportID, OriginAirportSeqID, OriginCityMarketID, OriginCityName,
    OriginState, OriginStateFips, OriginStateName, OriginWac, DestAirportID,
    DestAirportSeqID, DestCityMarketID, DestCityName, DestState, DestStateFips,
    DestStateName, DestWac, DepDel15, DepartureDelayGroups, DepTimeBlk, TaxiOut,
    WheelsOff, WheelsOn, TaxiIn, CRSArrTime, ArrDelay, ArrDel15, ArrivalDelayGroups,
    ArrTimeBlk, DistanceGroup, DivAirportLandings.
Thought 2: We need to filter the information related to the query.
Action 2: FilterDB[Flight_Number_Marketing_Airline=1575, FlightDate=2022-01-12,
    Origin=ATL, Dest=MCO]
Observation 2: We have successfully filtered the data (1 row).
Thought 3: We then need to know the departure delayed time.
Action 3: GetValue[DepDelay]
Observation 3: -7.0
Thought 4: We then need to know the arrival delayed time.
Action 4: GetValue[ArrDelay]
Observation 4: -17.0
Thought 5: To compute the extra minutes, we need to subtract the departure delayed
    time from the arrival delayed time.
Action 5: Calculate[(-17)-(-7)]
Observation 5: -10
```

```
Thought 6: After calculation, we know that the answer is -10.
Action 6: Finish[-10]

Question: Michael had 58 golf balls. On tuesday, he lost 23 golf balls. On
    wednesday, he lost 2 more. How many golf balls did he have at the end of
    wednesday?
Thought 1: Maybe I can write a python program to solve this problem.
Action 1: PythonInterpreter[# solution in Python:
def solution():
    # Michael had 58 golf balls. On tuesday, he lost 23 golf balls. On wednesday,
    # he lost 2 more. How many golf balls did he have at the end of wednesday?
    golf_balls_initial = 58
    golf_balls_lost_tuesday = 23
    golf_balls_lost_wednesday = 2
    golf_balls_left = golf_balls_initial - golf_balls_lost_tuesday - \
        golf_balls_lost_wednesday
    result = golf_balls_left
    return result]
Observation 1: 33
Thought 2: After execution of Python Code, we know that the answer is 33.
Action 2: Finish[33]

Question: What is the corresponding Mean_IoU score of the FRRN method on Cityscapes
    dataset for Semantic_Segmentation task?
Thought 1: The question is asking some performance information about Mean_IoU score
    of the FRRN method on Cityscapes dataset for Semantic_Segmentation task, we
    need to retrieve some useful information from the scirex database.
Action 1: RetrieveScirex[Mean_IoU score of the FRRN method on Cityscapes dataset
    for Semantic_Segmentation task]
Observation 1: The other stream undergoes a sequence of pooling operations to obtain
    robust features for recognition . The two streams are coupled at the full image
    resolution using residuals . Without additional processing steps and without
    pretraining , our approach achieves an intersection - over - union score of
    71.8 % on the Cityscapes dataset .
Thought 2: As the approach achieves an intersection - over - union score of 71.8 %
    on the Cityscapes dataset, the answer should be 71.8%.
Action 2: Finish[71.8%]

Question: When was the paper Learning the Principle of Least Action with
    Reinforcement Learning. published?
Thought 1: The question is asking some basic information of a dblp paper. We need
    to load the DBLP graph.
Action 1: LoadGraph[dblp]
Observation 1: DBLP graph is loaded.
Thought 2: The question is asking the published date of a paper, we need to check
    the node from the PaperNet in DBLP graph.
Action 2: NodeCheck[PaperNet, Learning the Principle of Least Action with
    Reinforcement Learning.]
Observation 2: {'title': 'Learning the Principle of Least Action with Reinforcement
    Learning.', 'year': 2021, 'venue': {'raw': 'AAAI Spring Symposium - MLPS'},
    'n_citation': 0, 'keywords': [], 'doc_type': 'Conference', 'page_start': '',
    'page_end': ''}
Thought 3: The published date of the paper is 2021.
Action 3: Finish[2021]

Question: How many collaborators does Chao Zhang have in the DBLP graph?
Thought 1: The question is asking the number of collaborators of a person, we
    need load the DBLP graph.
Action 1: LoadGraph[dblp]
Observation 1: DBLP graph is loaded.
Thought 2: The question is asking the collaborators of a person, we need to
    check the neighboring nodes from the AuthorNet in DBLP graph.
Action 2: NeighbourCheck[AuthorNet, Chao Zhang]
Observation 2: ['YUHUI YUAN', 'Rao Fu', 'Lang Huang', 'Weihong Lin', 'X Chen',
    'Jingdong Wang']
```

```
Thought 3: The number of collaborators of Chao Zhang is 6.
Action 3: Finish[6]

Question: How many papers does Chao Zhang and Weihong Lin have in common in the
    DBLP graph?
Thought 1: The question is asking the number of common papers of two persons, we
    need load the DBLP graph.
Action 1: LoadGraph[dblp]
Observation 1: DBLP graph is loaded.
Thought 2: The question is asking the common papers of two persons, we need to
    check the edges between them from the PaperNet in DBLP graph.
Action 2: EdgeCheck[PaperNet, Chao Zhang, Weihong Lin]
Observation 2: {'weight': 1, 'papers': ['HRFormer: High-Resolution Vision
    Transformer for Dense Predict.'], 'n_citation': [95]}
Thought 3: The number of common papers of Chao Zhang and Weihong Lin is 1.

Question: Where did Stephen's Opera performance take place?
Thought 1: The question is asking the location of Stephen's Opera performance
    from agenda.
Action 1: RetrieveAgenda[Stephen's Opera performance]
Observation 1: On January 29, 2022, there will be an opera performance at the Lyric
    Opera House, featuring Stephen. The show will start at 7:00 PM and end at
    9:00 PM. It promises to be a wonderful evening of beautiful music and powerful
    performances in a stunning venue. Come and experience the magic of opera at its
    finest!
Thought 2: The event happened in Lyric Opera.
Action 2: Finish[Lyric Opera]

Question: What was the trading volume of coffee on 2000-01-14?
Thought 1: Maybe I can write a SQL query to solve this problem.
Action 1: SQLInterpreter(SELECT Volume FROM coffee.coffee_data WHERE Date =
    '2000-01-14';
Observation 1: Volume: 10115
Thought 2: The volume of coffee on 2000-01-14 is 10115.
Action 2: Finish[10115]
```

The prompts used in Chameleon [32]:

<Chameleon> Prompt

```
You need to act as a policy model, that given a question and a modular set,
    determines the sequence of modules that can be executed sequentially can solve
    the question.

The modules are defined as follows:

- Calculate[formula]: This module calculates a given formula and returns the
    result. It takes in a mathematical formula and returns the calculated result.
    Normally, we only consider using "Calculate" when the question involves
    mathematical computations.

- RetrieveAgenda[keyword]: This module retrieves an agenda related to a specific
    keyword and returns it. It takes in a keyword and returns the corresponding
    agenda. Normally, we only consider using "RetrieveAgenda" when the question is
    about specific actions or tasks related to a topic.

- RetrieveScirex[keyword]: This module retrieves paragraphs from machine learning
    papers related to the specified keyword and returns them. It takes in a keyword
    and returns the relevant paragraphs. Normally, we only consider using
    "RetrieveScirex" when the question involves understanding specific concepts
    in machine learning.

- LoadDB[DBName]: This module loads a database specified by the database name and
    returns the loaded database. It takes in a database name and returns the
    corresponding database. The DBName can be one of the following: flights/
    coffee/airbnb/yelp. Normally, we only consider using "LoadDB" when the
    question requires data from a specific structured dataset.

- FilterDB[column_name, relation, value]: This module filters a database by a
    specified column name, relation, and value, and then returns the filtered
    database. It takes in a column name, a relation, and a value, and returns
    the filtered database. Normally, we only consider using "FilterDB" when the
    question requires a specific subset of data from a structured dataset.

- GetValue[column_name]: This module returns the value of a specified column in a
    database. It takes in a column name and returns its value. Normally, we only
    consider using "GetValue" when the question requires a specific piece of data
    from a structured dataset.

- LoadGraph[GraphName]: This module loads a graph specified by the graph name and
    returns the loaded graph. It takes in a graph name and returns the
    corresponding graph. Normally, we only consider using "LoadGraph" when the
    question involves understanding or navigating specific graph structures.

- NeighbourCheck[GraphName, Node]: This module lists the neighbors of a specified
    node in a graph and returns the neighbors. It takes in a graph name and a node,
    and returns the node's neighbors. Normally, we only consider using
    "NeighbourCheck" when the question involves understanding relationships in a
    graph structure.

- NodeCheck[GraphName, Node]: This module returns the detailed attribute
    information of a specified node in a graph. It takes in a graph name and a
    node, and returns the node's attributes. Normally, we only consider using
    "NodeCheck" when the question requires information about a specific entity in
    a graph.

- EdgeCheck[GraphName, Node1, Node2]: This module returns the detailed attribute
    information of the edge between two specified nodes in a graph. It takes in a
    graph name and two nodes, and returns the attributes of the edge between them.
    Normally, we only consider using "EdgeCheck" when the question involves
    understanding the relationship between two entities in a graph.

- SQLInterpreter[SQL]: This module interprets a SQL query and returns the result.
    It takes in a SQL query and returns the result of the query. Normally, we only
    consider using "SQLInterpreter" when the question requires data manipulation
    and extraction from a structured dataset.

- PythonInterpreter[Python]: This module interprets Python code and returns the
    result. It takes in Python code and returns the result of the code execution.
    Normally, we only consider using "PythonInterpreter" when the question requires
    complex computations or custom data manipulation.

- Finish[answer]: This module returns the final answer and finishes the task. This
    module is the final module in the sequence that encapsulates the result of all
    previous modules.

Below are some examples that map the problem to the modules.

Question: How many extra minutes did the DL1575 flight take from ATL to MCO on 2022-01-12?

Modules: ["LoadDB[flights]", "FilterDB[Flight_Number_Marketing_Airline=1575,
    FlightDate=2022-01-12, Origin=ATL, Dest=MCO]", "GetValue[DepDelay]",
    "GetValue[ArrDelay]", "Calculate[(-17)-(-7)]", "Finish[-10]"]

Question: Michael had 58 golf balls. On tuesday, he lost 23 golf balls. On
    wednesday, he lost 2 more. How many golf balls did he have at the end of
    wednesday?

Modules: ["PythonInterpreter[# solution in Python:\n\ndef solution():\n # Michael
    had 58 golf balls. On tuesday, he lost 23 golf balls. On wednesday, he lost 2
    more. How many golf balls did he have at the end of wednesday?\n
    golf_balls_initial = 58\n golf_balls_lost_tuesday = 23\n
    golf_balls_lost_wednesday = 2\n golf_balls_left =
    golf_balls_initial - golf_balls_lost_tuesday - golf_balls_lost_wednesday\n
    result = golf_balls_left\n return result]", "Finish[33]"]

Question: What is the corresponding Mean_IoU score of the FRRN method on Cityscapes
    dataset for Semantic_Segmentation task?

Modules: ["ScirexRetrieve[Mean_IoU score of the FRRN method on Cityscapes dataset
    for Semantic_Segmentation task]", "Finish[71.8%]"]

Question: When was the paper Learning the Principle of Least Action with
    Reinforcement Learning. published?

Modules: ["LoadGraph[dblp]", "NodeCheck[PaperNet, Learning the Principle of
    Least Action with Reinforcement Learning.]", "Finish[2021]"]

Question: How many collaborators does Chao Zhang have in the DBLP graph?

Modules: ["LoadGraph[dblp]", "NeighbourCheck[AuthorNet, Chao Zhang]", "Finish[6]"]

Question: How many papers does Chao Zhang and Weihong Lin have in common in
    the DBLP graph?

Modules: ["LoadGraph[dblp]", "EdgeCheck[PaperNet, Chao Zhang, Weihong Lin]",
    "Finish[1]"]

Question: Where did Stephen's Opera performance take place?

Modules: ["AgendaRetrieve[Stephen's Opera performance]", "Finish[Lyric Opera]"]

Question: What was the trading volume of coffee on 2000-01-14?

Modules: ["SQLInterpreter[SELECT Volume FROM coffee.coffee_data WHERE Date =
    '2000-01-14']", "Finish[10115]"]

Now, you need to act as a policy model, that given a question and a modular set,
    determines the sequence of modules that can be executed sequentially can
    solve the question.
```

G Key Information of ToolQA

G.1 Dataset Documentations

The dataset is provided in *jsonl* format. Each task corresponds to two files: easy and hard (*e.g.*, "flight-easy.jsonl" and "flight-hard.jsonl", *etc.*). Each data point contains the following fields:

- `qid`: the unique identifier for the question-answer pair;
- `question`: the question to query;
- `answer`: the corresponding ground-truth answer to `question`.

G.2 Intended Uses

ToolQA is intended for researchers in machine learning and related fields to innovate novel methods for tool-augmented large language models (LLMs). We also aim to help developers to test their plugins on our dataset.

G.3 Hosting and Maintenance Plan

ToolQA codebase is hosted and version-tracked via GitHub. It will be permanently available under the link https://github.com/night-chen/ToolQA. The download link of all the datasets can be found in the GitHub repository.

ToolQA is a community-driven and open-source initiative. We are committed and have resources to maintain and actively develop ToolQA in the future. We plan to grow ToolQA to include more tasks, tools, and more baseline methods. We welcome external contributors.

G.4 Licensing

We license our work using Apache 2.0[12]. All the datasets will be publicly released through the aforementioned GitHub link.

G.5 Statement

The authors will bear all responsibility in case of violation of rights.

[12] https://www.apache.org/licenses/LICENSE-2.0