Large Language Model Applications

Tool-Use and Chatbots

Daphne Ippolito and Fernando Diaz

Tool-Use

Intent detection followed by slot-filling

- Chatbot first figures out what the user's intent is.
  - `"Are you looking to book a new flight or modify an existing reservation?"`
- Once the chatbot knows the task, it can ask questions to fill in the set of information that, once collected, allows the task to be completed autonomously.
  - `"Where would you like to fly?"`
  - `"What are your departure and return dates?"`

An alternative method: tool use

- **Tool**: an API call that can perform actions and return results
- **General idea**: Conversational LLM is trained to generate API calls when appropriate, and incorporate their outputs into its responses.

Citations

Difference from prior methods

- More flexible than intent detection and slot filling. We no longer need to explicitly model user intent or slots to fill.
    - LLM simply generates code, the same as it generates text.
- In the past, companies would implement task-oriented chatbots for specific purposes. Chatbot takes action when all slots are filled.
    - `An airline has a chatbot that can make and modify flight bookings.`
    - `A restaurant has a chatbot that can help you to order pizza.`
- Tool-use models are still general-purpose conversational language models, rather than single-purpose task completers.

Citations

What can tools do?

Category	Example Tools

Knowledge access	`sql_executor(query: str) -> answer: any`
	`search_engine(query: str) -> document: str`
	`retriever(query: str) -> document: str`

Computation activities	`calculator(formula: str) -> value: int | float`
	`python_interpreter(program: str) -> result: any`
	`worksheet.insert_row(row: list, index: int) -> None`

Interaction w/ the world	`get_weather(city_name: str) -> weather: str`
	`get_location(ip: str) -> location: str`
	`calendar.fetch_events(date: str) -> events: list`
	`email.verify(address: str) -> result: bool`

Non-textual modalities	`cat_image.delete(image_id: str) -> None`
	`spotify.play_music(name: str) -> None`
	`visual_qa(query: str, image: Image) -> answer: str`

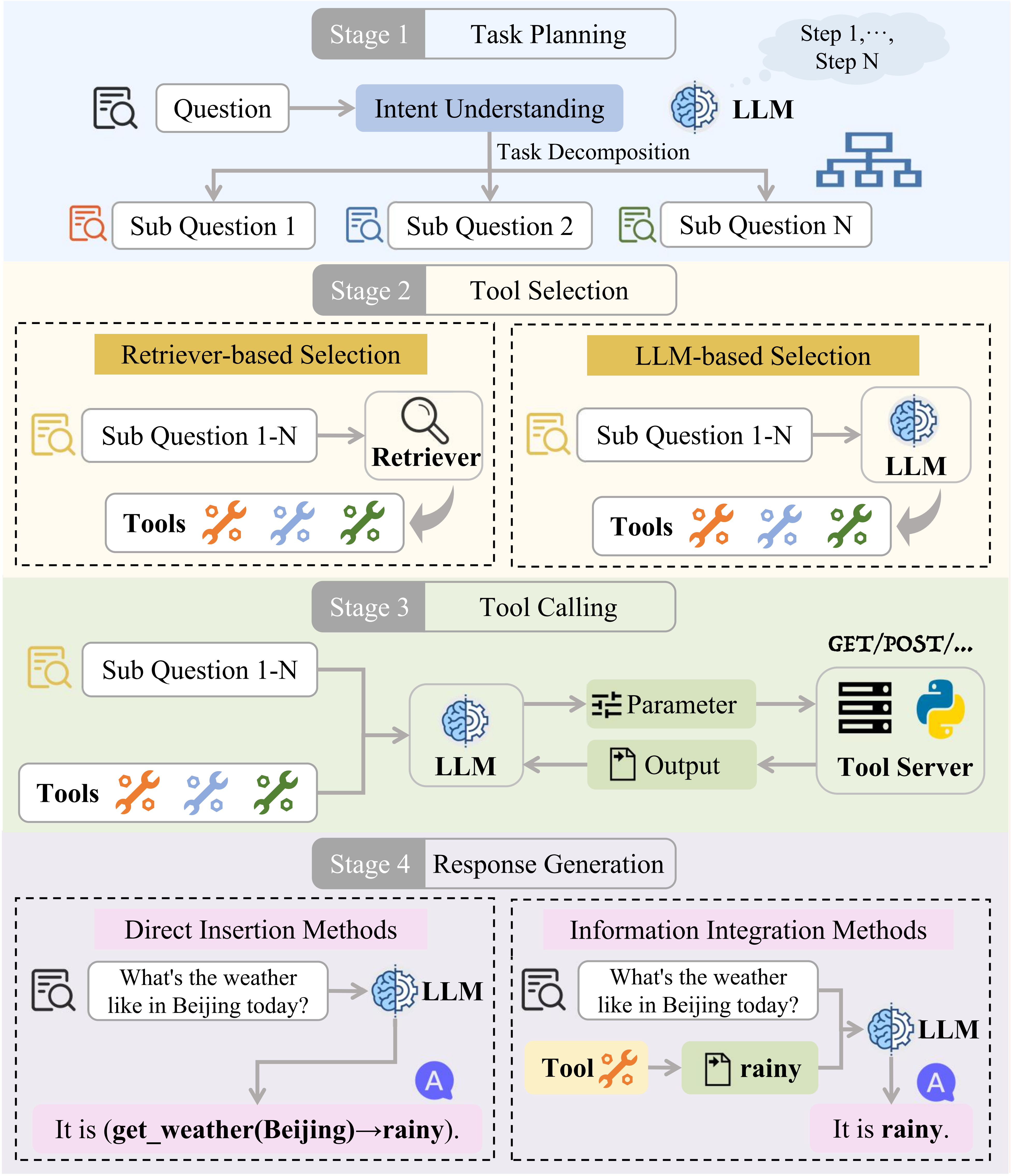
Special-skilled LMs	`QA(question: str) -> answer: str` `translation(text: str, language: str) -> text: str`

Wang et al. "What Are Tools Anyway? A Survey from the Language Model Perspective." 2024.

Properties we want from a general-purpose tool-calling AI

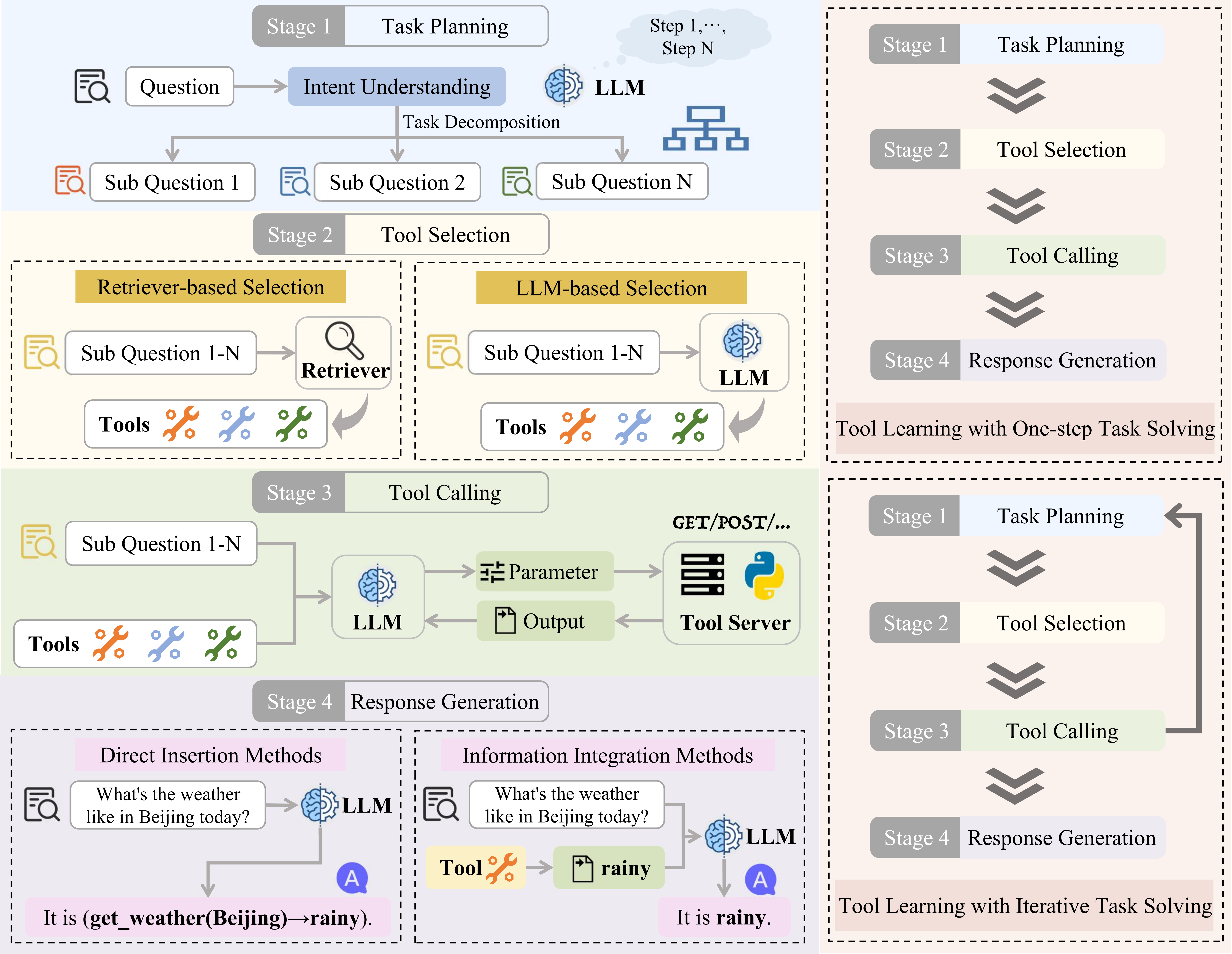
1. AI should have access to dozens or even hundreds of tools
2. AI should generalize to tools not seen during training
3. AI chooses when during the generation process to make a tool call
4. AI has the ability to make multiple tool calls when necessary

Typical tool-use flows circa 2024



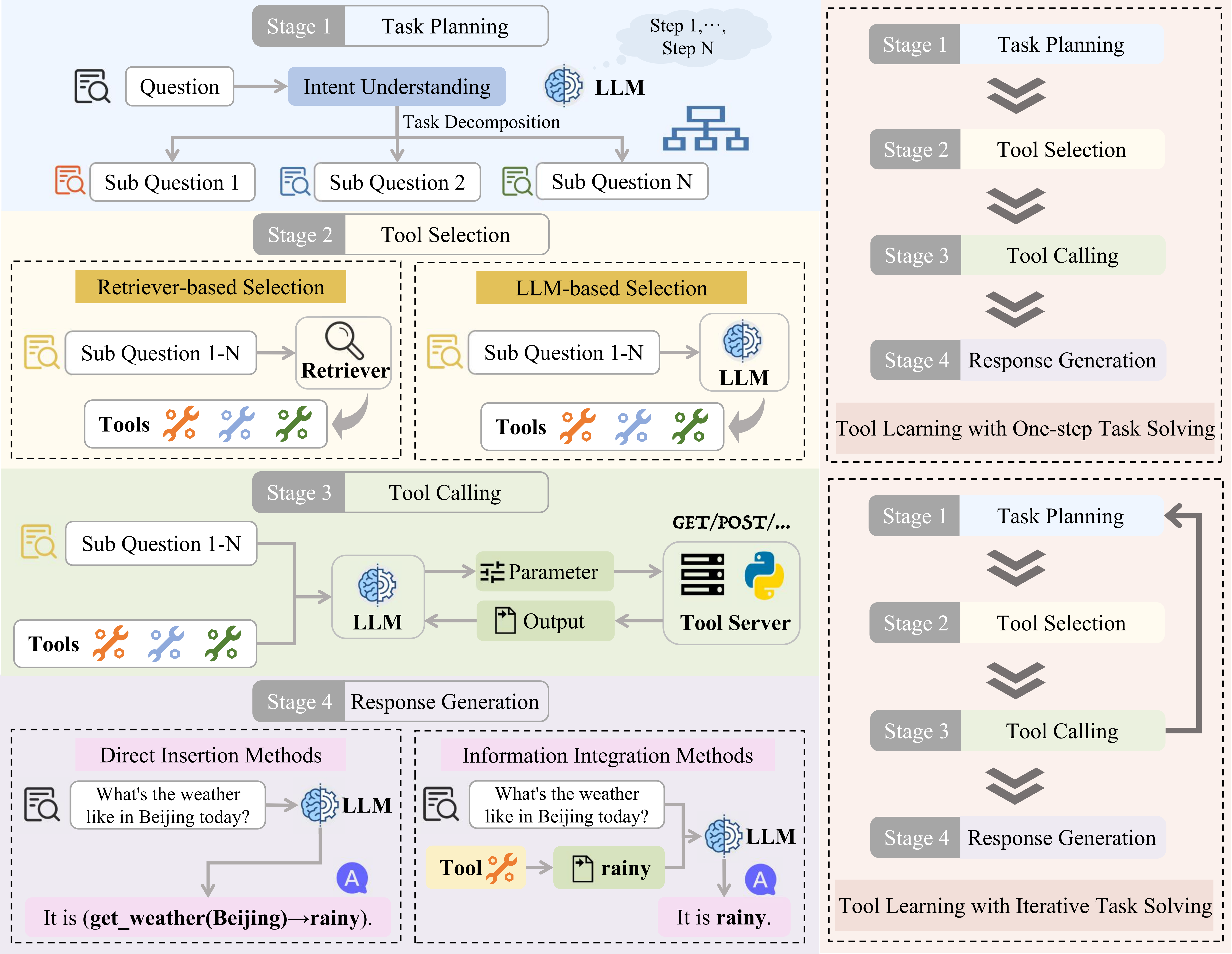
Qu et al. "Tool Learning with Large Language Models: A Survey." 2024.

Typical tool-use flows circa 2024



Qu et al. "Tool Learning with Large Language Models: A Survey." 2024.

Typical tool-use flows circa 2024

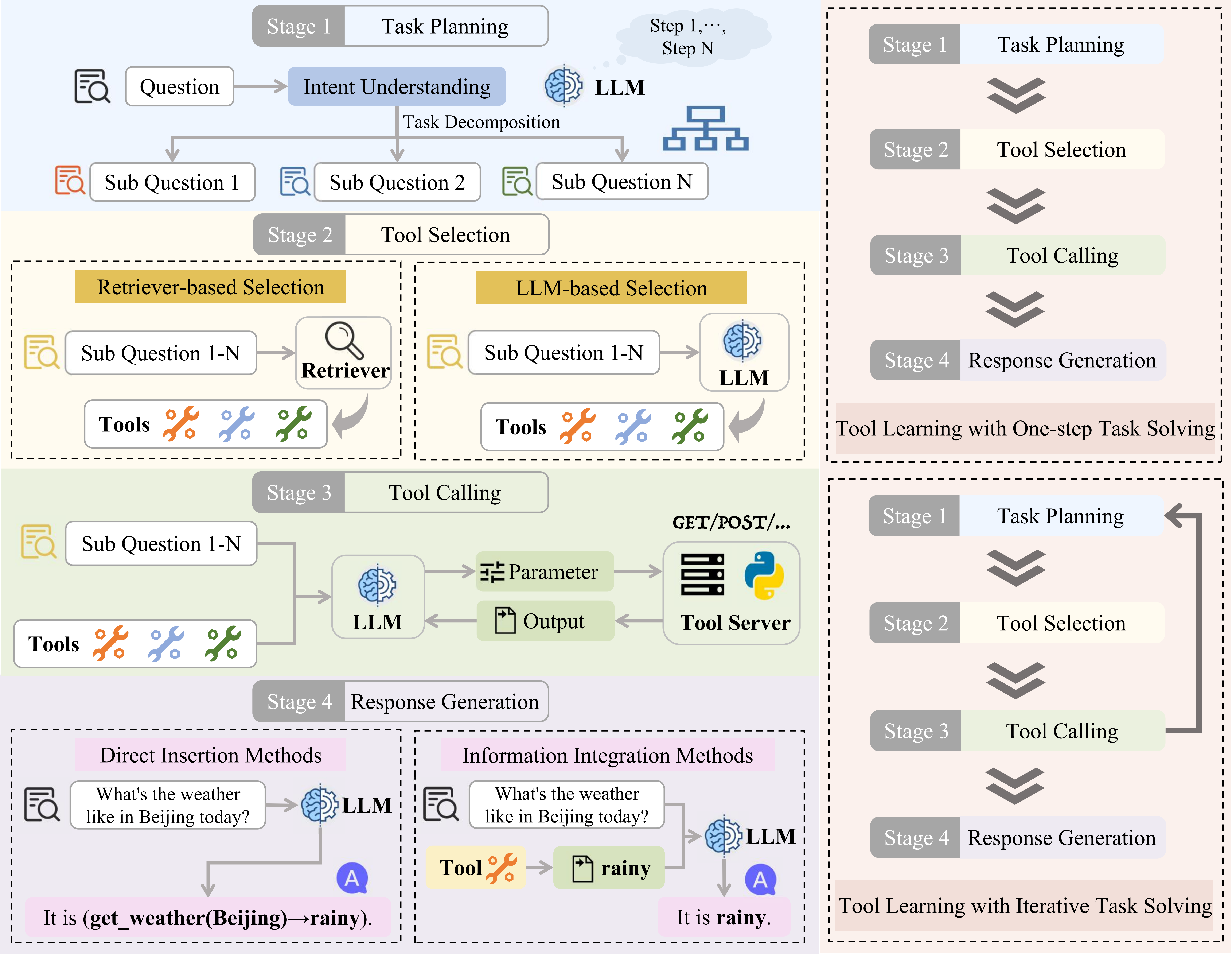


Desired properties:

1. AI should have access to dozens or even hundreds of tools
2. AI should generalize to tools not seen during training
3. AI chooses when during the generation process to make a tool call
4. AI has the ability to make multiple tool calls when necessary

Qu et al. “Tool Learning with Large Language Models: A Survey.” 2024.

Typical tool-use flows circa 2024



2026:

- Task planning is increasingly merged with reasoning
- Retriever-based selection followed by LLM-based selection from top retrieved tools is the norm
- Iterative tasking solving means [reasoning→tool call→reasoning→tool call→etc.]
- During response generation, LLM acts on tool's output.

Qu et al. "Tool Learning with Large Language Models: A Survey." 2024.

Two case studies

- Toolformer
- ToolBench

Citations

Toolformer

- Took an existing LM dataset, and annotated it with tool calls.
- Finetuned a pre-trained LLM on the annotated data.

The New England Journal of Medicine is a registered trademark of [QA("Who is the publisher of The New England Journal of Medicine?") → Massachusetts Medical Society] the MMS.

Out of 1400 participants, 400 (or [Calculator(400 / 1400) → 0.29] 29%) passed the test.

The name derives from "la tortuga", the Spanish word for [MT("tortuga") → turtle] turtle.

The Brown Act is California's law [WikiSearch("Brown Act") → The Ralph M. Brown Act is an act of the California State Legislature that guarantees the public's right to attend and participate in meetings of local legislative bodies.] that requires legislative bodies, like city councils, to hold their meetings open to the public.

- At generation time, model generates tool calls.

Schick et al. "Toolformer: Language Models Can Teach Themselves to Use Tools." 2023.

Toolformer

- Only a limited number of tools supported.

API Name	Example Input	Example Output
Question Answering	Where was the Knights of Columbus founded?	New Haven, Connecticut
Wikipedia Search	Fishing Reel Types	Spin fishing > Spin fishing is distinguished between fly fishing and bait cast fishing by the type of rod and reel used. There are two types of reels used when spin fishing, the open faced reel and the closed faced reel.
Calculator	27 + 4 * 2	35
Calendar	ε	Today is Monday, January 30, 2023.
Machine Translation	sûreté nucléaire	nuclear safety

- Takeaways:
  - Tool calls improved performance at knowledge and math benchmarks.
  - Synthesized tool-call data makes good training data.

Schick et al. "Toolformer: Language Models Can Teach Themselves to Use Tools." 2023.

ToolBench

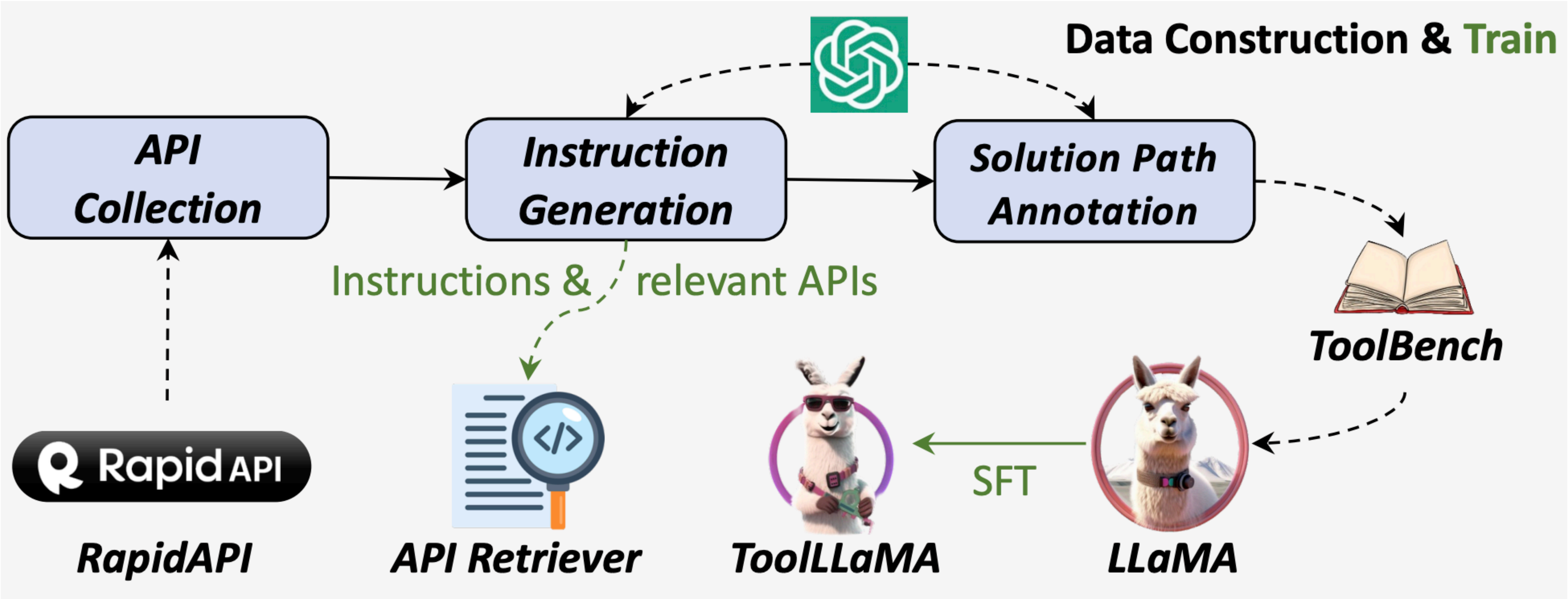
an instruction-tuning dataset for training tool use models

- Dataset of 88.9k reasoning+tool-use trajectories.
  - `User input: I would like to explore different fish species. Can you provide me with a list of available fish species and their images?`
  - `Call /fish_api/fishes (api_description="This endpoint will return back all available fishes that are available", required_parameters=[])`
  - `Call /fish_api/fish/{name} (api_description="This endpoint will return information for a specific fish", required_parameters=["name"])`
  - `Target output: "`
- Dataset created by prompting GPT-4, then used to fine-tune LLaMa for tool calling.

Qin et al. "ToolLLM: Facilitating Large Language Models to Master 16000+ Real-world APIs." 2023.

ToolBench

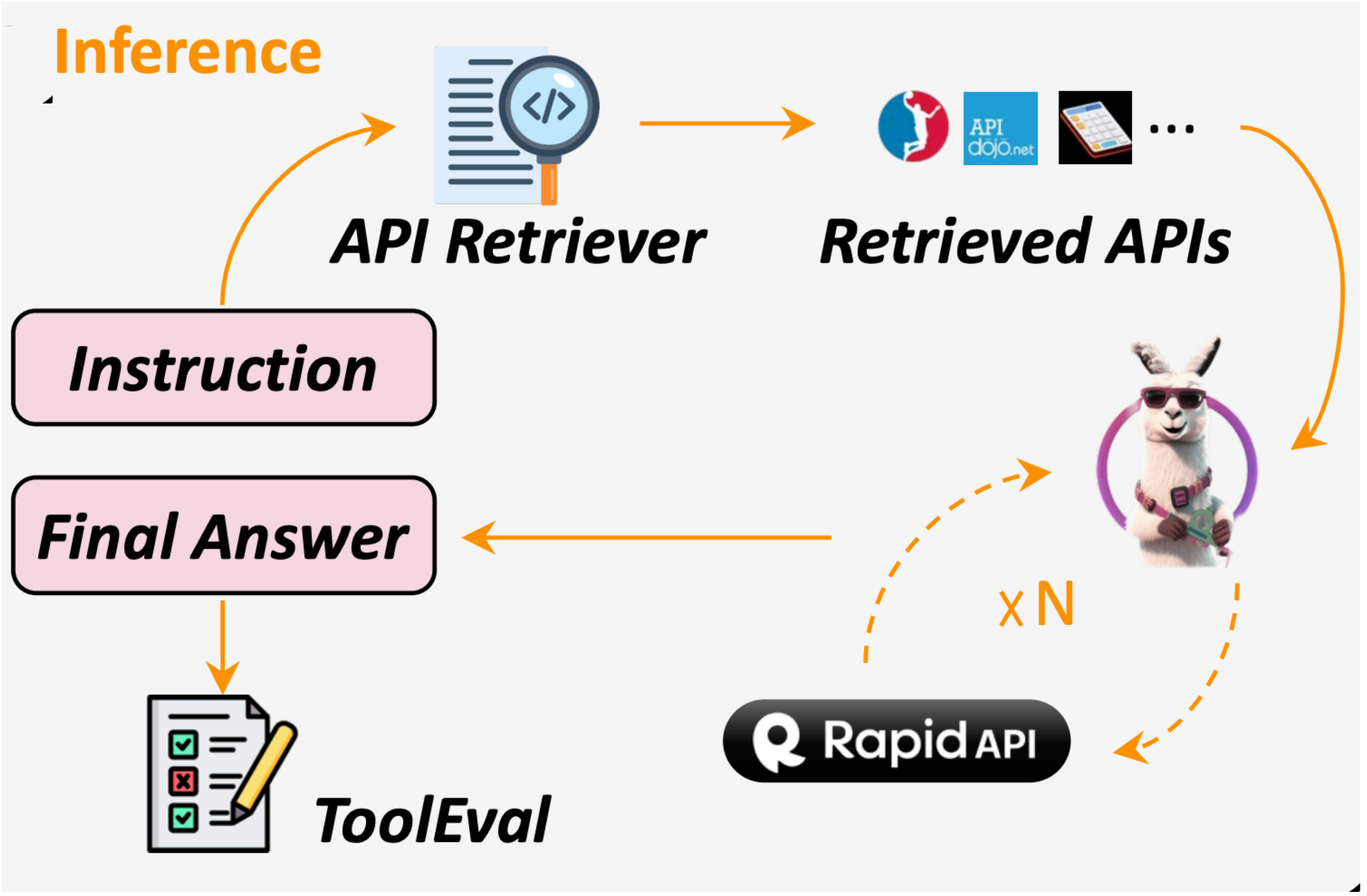
an instruction-tuning dataset for training tool use models



Qin et al. "ToolLLM: Facilitating Large Language Models to Master 16000+ Real-world APIs." 2023.

ToolBench

an instruction-tuning dataset for training tool use models



Qin et al. "ToolLLM: Facilitating Large Language Models to Master 16000+ Real-world APIs." 2023.

Evaluating tool-use

- Does incorporating the tool's output into the AI's decision-making increase the chance it arrives at the correct answer?
- Was the correct environmental action taken?
- What are the costs of tool-use (e.g. increased latency, inference costs if tool is another LLM)?

Risks of tools

- TODO

Model Context Protocol

Suppose you've built a robot that wanders around Gates-Hillman according to instructions students send out.

You want all students—regardless of whether they use Claude, or Gemini, or GPT-5—to be able to send instructions to the robot.

Model Context Protocol

- **Host:** The AI application that coordinates and manages one or multiple MCP clients
- **Client:** A component that maintains a connection to an MCP server and obtains context from an MCP server for the MCP host to use
- **Server:** A program that provides context to MCP clients

Model Context Protocol

Server

- Server declares its set of capabilities:
  - Tools: functions that the client model can actively call
  - Resources: read-only information the AI application can retrieve and provide as context to the client model.
  - Prompts: pre-built instruction templates that provide expected interaction patterns; telling model how to work with specific tools and resources

Model Context Protocol

Client

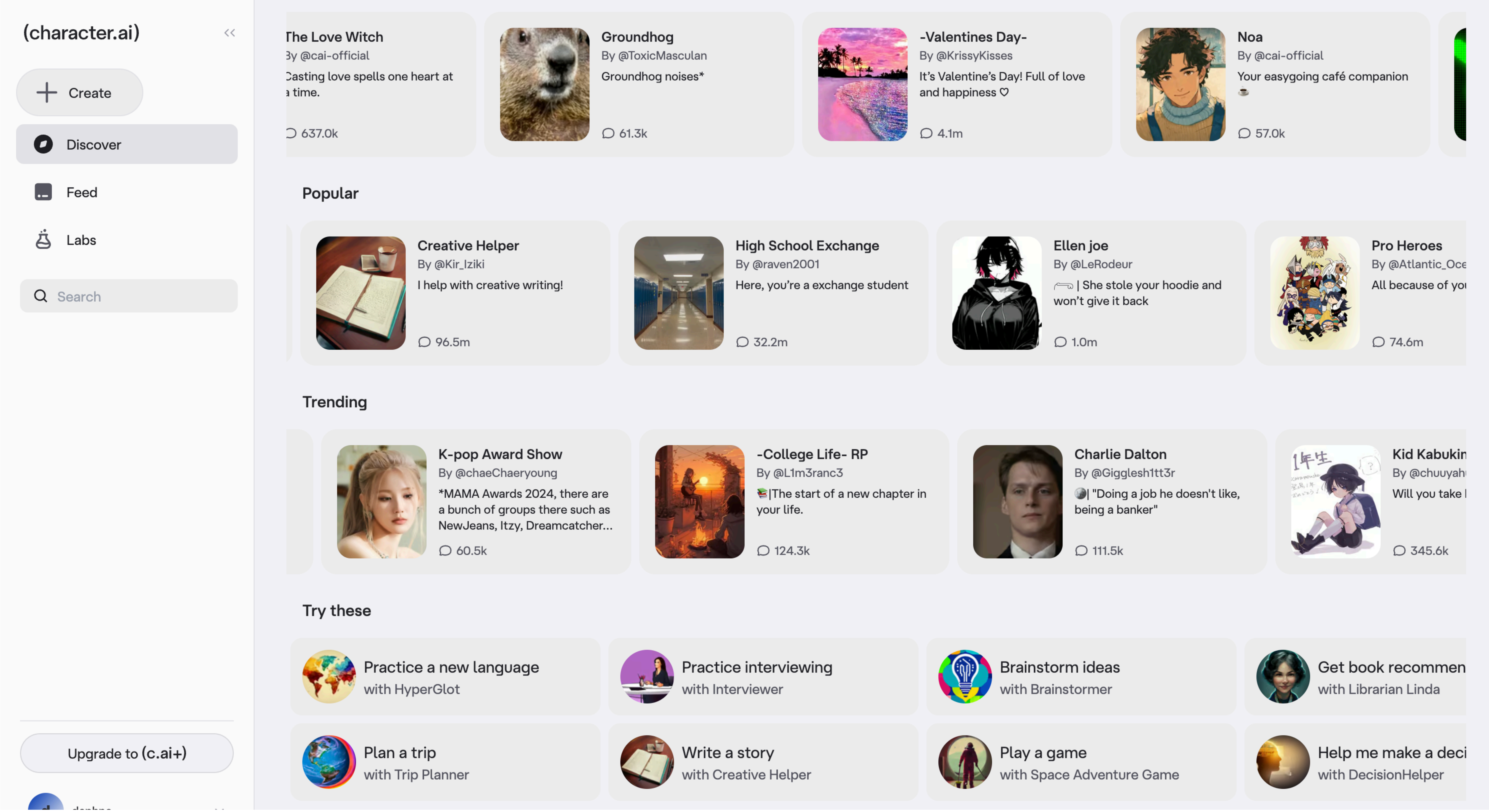
- Processes user queries by
  - Calling an LLM
  - Calling tools

Personas and Companionship

How should a chatbot answer?

Should chatbots have personas?

Applications character.ai



Citations

Applications Tavern AI

Sally Tavern AI Characters list

Sally Tavern AI has three preset roles in the initial state, which are:

Coding Sensei

Coding Sensei is a helpful entity designed to assist users with their coding queries. It not only answers questions but also provides example codes, ensuring they are presented clearly within markdown codeblocks for optimal understanding and implementation.

Flux the Cat

Flux the Cat is a smart, cool, and quick-witted feline with a unique penchant for riding a Gundam-like Roomba. With a mix of black and white fur, yellow eyes, and a fluffy tail, Flux is wary yet well-trained, performing tricks for treats. While he loves his Roomba, cat treats, meats, and gazing at birds, he dislikes vegetables, bad smells, and interruptions during his naps. Flux communicates through typical cat noises and has clear boundaries with the user, ensuring a respectful and entertaining.

Seraphina
Eldoria, ch
compassic
nurturing c
with the fo
Physically,
amber eye
sundress.
in the vine
In the give
rescuing y
and care ir

What is Tavern AI?

Tavern AI is an advanced AI chatbot platform that offers an engaging conversationa
utilizes powerful language models capable of comprehending human input, includin
a variety of characters, each with distinct definitions and personalities. This enables
to their preferences.

Tavern AI Features

Tavern AI offers a range of features that enhance the user experience:

- Character Building: Users can create AI-based characters infused with specific motivation.
- Multiple Language Model Support: The platform supports a range of APIs, incl and Pygmalion.
- Customizable User Interface: Users can personalize their chat experience by a background, UI colors, chat size, avatar styles, and more.
- Immersive Chat Experience (Story Mode): Tavern AI offers an immersive and i actively participate in dynamic conversations with users.
- Group Chats: Users can simultaneously engage in conversations with multiple

Citations

How to give a chatbot a persona

- Prompting
- Finetuning
- Steering vectors

Citations

Prompting to create a persona

Silly Tavern, character.ai etc.—each chatbot has a “character card”

```
type TavernCardV2 = {
  spec: 'chara_card_v2'
  spec_version: '2.0' // May 8th addition
  data: {
    name: string
    description: string
    personality: string
    scenario: string
    first_mes: string
    mes_example: string
    creator_notes: string
    system_prompt: string
    post_history_instructions: string
    alternate_greetings: Array<string>
    character_book?: CharacterBook
    tags: Array<string>
    creator: string
    character_version: string
    extensions: Record<string, any>
  }
}
```

https://github.com/malfoyslastname/character-card-spec-v2

Prompting to create a persona

Silly Tavern, character.ai etc.—each chatbot has a "character card"

```
type TavernCardV2 = {
  spec: 'chara_card_v2'
  spec_version: '2.0' // May 8th addition
  data: {
    name: string
    description: string
    personality: string
    scenario: string
    first_mes: string # The first utterance the chatbot says
    mes_example: string
    creator_notes: string
    system_prompt: string
    post_history_instructions: string
    alternate_greetings: Array<string>
    character_book?: CharacterBook
    tags: Array<string>
    creator: string
    character_version: string
    extensions: Record<string, any>
  }
}
```

https://github.com/malfoyslastname/character-card-spec-v2

Prompting to create a persona

Silly Tavern, character.ai etc.—each chatbot has a "character card"

```
type TavernCardV2 = {
  spec: 'chara_card_v2'
  spec_version: '2.0' // May 8th addition
  data: {
    name: string
    description: string
    personality: string
    scenario: string
    first_mes: string
    mes_example: string
    creator_notes: string
    system_prompt: string # Instructions for the LM
    post_history_instructions: string
    alternate_greetings: Array<string>
    character_book?: CharacterBook
    tags: Array<string>
    creator: string
    character_version: string
    extensions: Record<string, any>
  }
}
```

Write {{char}}'s next reply in a fictional chat between {{char}} and {{user}}. Write 1 reply only in internet RP style, italicize actions, and avoid quotation marks. Use markdown. Be proactive, creative, and drive the plot and conversation forward. Write at least 1 paragraph, up to 4. Always stay in character and avoid repetition.

https://github.com/malfoyslastname/character-card-spec-v2

Prompting to create a persona

Silly Tavern, character.ai etc.—each chatbot has a “character card”

```
type TavernCardV2 = {
  spec: 'chara_card_v2'
  spec_version: '2.0' // May 8th addition
  data: {
    name: string
    description: string
    personality: string
    scenario: string
    first_mes: string
    mes_example: string
    creator_notes: string
    system_prompt: string
    post_history_instructions: string # Extra “system instructions” to put at end of prompt
    alternate_greetings: Array<string>
    character_book?: CharacterBook
    tags: Array<string>
    creator: string
    character_version: string
    extensions: Record<string, any>
  }
}
```

https://github.com/malfoyslastname/character-card-spec-v2

Prompting to create a persona

Silly Tavern, character.ai etc.—each chatbot has a "character card"

```
type TavernCardV2 = {
  spec: 'chara_card_v2'
  spec_version: '2.0' // May 8th addition
  data: {
    name: string
    description: string
    personality: string
    scenario: string
    first_mes: string
    mes_example: string
    creator_notes: string
    system_prompt: string
    post_history_instructions: string
    alternate_greetings: Array<string>
    character_book?: CharacterBook # Stores information about the world the bot inhabits.
    tags: Array<string>
    creator: string
    character_version: string
    extensions: Record<string, any>
  }
}
```

https://github.com/malfoyslastname/character-card-spec-v2

Challenges with these approaches

Finetuning-based personas

Steering vectors

Anthropic's persona vectors

- A persona vector v_ℓ is a vector that when added to the model's activations at layer ℓ, steers behavior toward some direction: $h_\ell \leftarrow h_\ell + \alpha \cdot v_\ell$

Weng et al. "ControlLM: Crafting Diverse Personalities for Language Models." 2024.

Steering vectors

Anthropic's persona vectors

- A persona vector v_ℓ is a vector that when added to the model's activations at layer ℓ, steers behavior toward some direction: $h_\ell \leftarrow h_\ell + \alpha \cdot v_\ell$
- How it's computed:
  1. Prompt Claude to have personality P, and collect all responses that strongly exhibit this personality.
  2. Prompt Claude to have personality $\neg P$, and do the same.
  3. In both cases, extract residual stream activations at every layer and average across response tokens.
  4. Compute the persona vector as the difference in mean activations between responses that exhibit the personality trait and those that do not.

Weng et al. "ControlLM: Crafting Diverse Personalities for Language Models." 2024.

Ways to use steering vectors

- At inference time, apply (or else subtract) the steering vector to guide generation's characteristics. Researchers found this to degrade general capabilities.
- At finetuning time, add steering vector in order to limit personality trait shifts.
  - TODO more here

Citations

Do you think that

Dangers in Chatbots