mini007 - A Lightweight Framework for Multi-Agents Orchestrations in R

R+AI 2025

Mohamed El Fodil Ihaddaden, Analytics Engineer, HDI Global SE

Few words about me

- Analytics Engineer at HDI Global 
- From Algiers, Algeria  Based in Hamburg, Germany 
- Open source R developer (more than 20 packages on CRAN)

mini007

- R package for multi-LLM-agents workflows
- Built on top of ellmer

- Available on CRAN

Features

- Agents memory and context management

Features

- Agents memory and context management
- Cost/Usage planning and monitoring

Features

- Agents memory and context management
- Cost/Usage planning and monitoring
- Simplicity and flexibility of usage

Features

- Agents memory and context management
- Cost/Usage planning and monitoring
- Simplicity and flexibility of usage
- Agent-to-Agent orchestration & result chaining

Features

- Agents memory and context management
- Cost/Usage planning and monitoring
- Simplicity and flexibility of usage
- Agent-to-Agent orchestration & result chaining
- Human In The Loop

Core Concepts

- **Agent**: An extension of ellmer but more flexible and powerful

Core Concepts

- **Agent**: An extension of *ellmer* with more flexibility

- **Lead Agent**: orchestrator, organize the LLM workflow

Core Concepts

- **Agent**: An extension of *ellmer* with more flexibility

- **Lead Agent**: orchestrator, organize the LLM workflow

Agent Usage

Creating an Agent

```r
library(mini007)

openai_4_1_mini <- ellmer::chat(
  name = "openai/gpt-4.1-mini",
  api_key = Sys.getenv("OPENAI_API_KEY"),
  echo = "none"
)

polar_bear_researcher <- Agent$new(
  name = "POLAR BEAR RESEARCHER",
  instruction = "You are an expert in polar bears, you task is to collect
information about polar bears. Answer in 1 sentence max.",
  llm_object = openai_4_1_mini
)
```

Invoking the Agent

```
polar_bear_researcher$invoke("Who are you?")

[1] "I am an expert in polar bears, dedicated to sharing knowledge about
their biology, habitat, and behavior."

polar_bear_researcher$invoke("Are polar bears dangerous?")

[1] "Yes, polar bears are dangerous wild animals and can be aggressive toward
humans if encountered."

polar_bear_researcher$invoke("Where can I find polar bears?")

[1] "Polar bears are primarily found in the Arctic region, including
countries like Canada, Alaska (USA), Greenland, Norway, and Russia."
```

Metadata

```
polar_bear_researcher$name
[1] "POLAR BEAR RESEARCHER"

polar_bear_researcher$agent_id
[1] "061bdf51-b991-43a0-ac31-84c85f34c16e"

polar_bear_researcher$model_name
[1] "gpt-4.1-mini"

polar_bear_researcher$model_provider
[1] "OpenAI"
```

Basic Usage: Access to ellmer

```
# Accessing the underlying ellmer object
polar_bear_researcher$llm_object
```

Budgeting

```
polar_bear_researcher$set_budget(0.0001)
✔ Budget successfully set to 1e-04$
ℹ Budget policy: on_exceed='abort', warn_at=0.8
ℹ Use the set_budget_policy() method to configure the budget policy.


polar_bear_researcher$invoke("Do you think polar bears are cute?")
Error in `private$.check_budget()` at mini007/R/Agent.R:102:9:
! POLAR BEAR RESEARCHER agent has exceeded its budget.
Cost: 5e-04, Budget: 1e-04
```

Managing messages

```
# Accessing the messages
polar_bear_researcher$messages

# Keeping only the last n messages (The system prompt is kept by default)
polar_bear_researcher$keep_last_n_messages(2)

# Clearing the messages history while summarising it into the system prompt
polar_bear_researcher$clear_and_summarise_messages()
✔ Conversation history summarised and appended to system prompt.
ℹ Summary: The user asked if polar bears are cute, and the expert assistant
responded that polar bears, particu...

# Exporting messages history
polar_bear_researcher$export_messages_history()
✔ Conversation saved to RESEARCHER_messages.json

# Loading messages history
polar_bear_researcher$load_messages_history()
✔ Conversation history loaded from POLAR BEAR RESEARCHER_messages.json
```

Generating/Executing R code

```r
polar_bear_researcher$generate_execute_r_code(
  code_description = glue::glue(
    "Choose 2 variables from the palmerpenguins dataset
    and run a linear regression"
  ),
  validate = TRUE,
  execute = TRUE,
  interactive = FALSE
)
```

Generating/Executing R code

```
ℹ Executing generated R code...
✔ Code executed successfully
$description
Choose 2 variables from the palmerpenguins dataset
and run a linear regression

$code
[1] "library(palmerpenguins);data<-na.omit(penguins);lm_result<-
lm(body_mass_g~flipper_length_mm,data=data);summary(lm_result)"

$validated
[1] TRUE

$validation_message
[1] "Syntax is valid"

$executed
[1] TRUE

$execution_result
$execution_result$value

Call:
lm(formula = body_mass_g ~ flipper_length_mm, data = data)

Residuals:
     Min       1Q   Median       3Q      Max
-1057.33  -259.79   -12.24   242.97  1293.89

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)     -5872.09     310.29  -18.93   <2e-16 ***
flipper_length_mm    50.15       1.54   32.56   <2e-16 ***
---
Signif. codes:
0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 393.3 on 331 degrees of freedom
Multiple R-squared:  0.7621,    Adjusted R-squared:  0.7614
F-statistic:  1060 on 1 and 331 DF,  p-value: < 2.2e-16


$execution_result$output
character(0)
```

Lead Agent Usage

Creating the necessary agents

```
researcher <- Agent$new(
  name = "researcher",
  instruction = "You are a research assistant. Your job is to answer factual
questions with detailed and accurate information. Do not answer with more
than 2 lines",
  llm_object = openai_4_1_mini
)

summarizer <- Agent$new(
  name = "summarizer",
  instruction = "You are agent designed to summarise a give text into 3
distinct bullet points.",
  llm_object = openai_4_1_mini
)

translator <- Agent$new(
  name = "translator",
  instruction = "Your role is to translate a text from English to German",
  llm_object = openai_4_1_mini
)
```

Registering the Agents

```
lead_agent <- LeadAgent$new(
  name = "Leader",
  llm_object = openai_4_1_mini
)

lead_agent$register_agents(c(researcher, summarizer, translator))
✔ Agent(s) successfully registered.
```

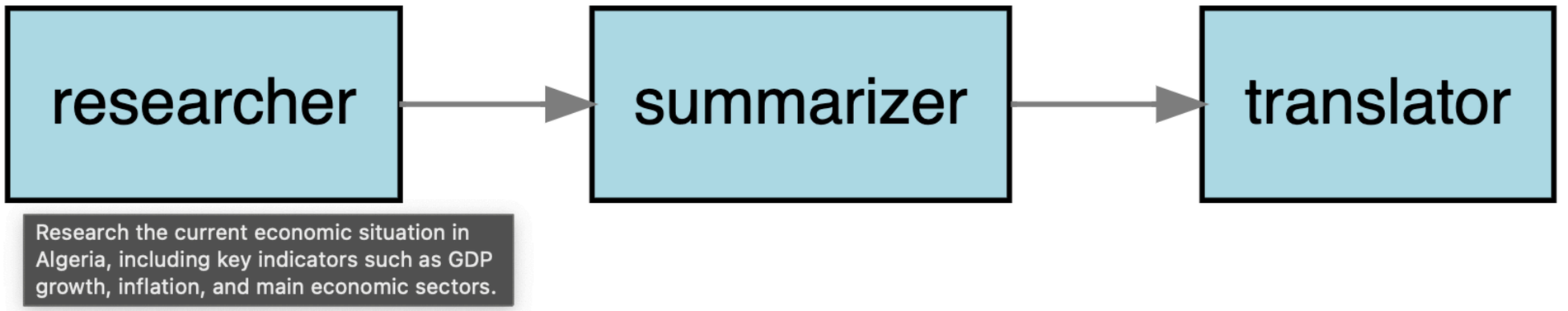
Generating a plan

```
prompt_to_execute <- "Tell me about the economic situation in Algeria,
summarize it in 3 bullet points, then translate it into German."

lead_agent$generate_plan(prompt_to_execute)
```

Visualizing the plan

```
lead_agent$visualize_plan()
```



Executing the workflow

```
lead_agent$invoke("Tell me about the economic situation in Algeria, summarize
it in 3 bullet points, then translate it into German.")
```

Human In The Loop (HITL)

```
lead_agent$set_hitl(steps = c(1, 2))
✔ HITL successfully set at step(s) 1, 2.
```

Broadcasting

```
openai_4_1_agent <- Agent$new(
  name = "openai_4_1_agent",
  instruction = "You are an AI assistant. Answer in 1 sentence max.",
  llm_object = ellmer::chat(
    name = "openai/gpt-4.1",
    api_key = Sys.getenv("OPENAI_API_KEY"),
    echo = "none"
    )
)

openai_4_1_nano_agent <- Agent$new(
  name = "openai_4_1_nano_agent",
  instruction = "You are an AI assistant. Answer in 1 sentence max.",
  llm_object = ellmer::chat(
    name = "openai/gpt-4.1-nano",
    api_key = Sys.getenv("OPENAI_API_KEY"),
    echo = "none"
    )
)
```

Broadcasting

```
lead_agent$register_agents(c(openai_4_1_agent, openai_4_1_nano_agent))
✔ Agent(s) successfully registered.

lead_agent$broadcast(
  prompt = "Sing me a song about an Algerian poet enjoying the rain"
)
```

Judge as a decision process

```
lead_agent$judge_and_choose_best_response(
  "what's the best way to wear a blue kalvin klein shirt in winter with a
pink pair of trousers?"
)
```

Thank you