

Keynote
Steps: Prompt > String > Function
Steps: Human request > LLM response > Human call functions
*request = natural language query, i.e. text, specification can be converted into function *response = string
String
- description of a function - function name - when to use it
Function = tools
- extend existing tool's function - build new function, i.e. AI agent
LLM function calling
- LLM take in natural language query, i.e. description of a function, function name, parameter details - output a string for human to call a function
Note
- LLMs do not call functions directly - return a string to call a function
Code
- set up raven prompt/ user query - set up raven call from utils import query_raven call = query_raven(raven_prompt) print (call)
- print (raven_call): LLM return string - exec (raven_call): human call function
LLM function calling types
- single: single query + single defined function - multiple: single query + multiple defined functions - parallel: multiple queries (execute different parent functions in the same turn) + single/ multiple defined functions - nested: multiple queries (execute all sister functions in one step) + multiple defined functions - no call
LLM function calling use cases
- <u>condition 1: execute a task</u> - steps: (API interfacing) > set up user query > define function > set up raven call > print (raven_call) > exec (raven_call)
- <u>condition 2: extract structured data from unstructured data</u> - steps: set up raven prompt > define function > set up raven call > eval (raven_prompt)
- <u>condition 3: generate valid JSON</u> - steps: set up raven prompt > define function > set up user query > eval (json) > print (json)
- <u>condition 4: retrieve current data from web to self-learn and update</u> - steps: set up raven prompt > define function (add example to show how to search from web) > print (raven_prompt)
- <u>condition 5: retrieve insights from internal database</u> - internal database is not accessed by public - need to ask LLM to learn SQL to retrieve the insight for a request/ user query - steps: set up raven prompt > define function (define sql code, schema) > print (raven_prompt)
*unstructured data = natural language input *steps are FYI
Parent functions: independent function
- Draw a clown face - Draw a tie
Sister functions: associated functions in a parent function
- Draw eyes - Draw noses
Large language model > big model > pre-trained > general purpose model > execute all the queries/ requests Small language model > small model > fine-tuned > special purpose model > execute small set of queries/ requests
- LLM examples: GPT, Gemini - SLM examples: Nexusflow
Small language model
- can be hosted locally - prevent latency - save cost

## AI agent

- understand human prompts
- can self-learn a string from a prompt
- take action according to the prompt

\*prompt has function definitions, i.e. parameter details

## AI agent example

- analyze user feedback
- extract useful feedback by SQL
- help store in database

With defined functions, you can call and execute all/ part/ none of them

## Function calling fail

- add a doc string to add more descriptions of a function in parameter section
- re-print (raven\_call)
- exec (raven\_call)

Raven call can be used to interfaced with external tools by API

## Structured extraction (data)

- Method 1 - simple method for balanced data
- Method 2 - data class method for imbalanced data, i.e. define to-be-extracted data one by one in prompt, with data type and add. note

## Course project: Dialogue Data Extraction System

Goal: extract key data from dialogue between customers and customer representatives by LLM

Required data to be extracted: agent name, customer data, i.e. customer name, phone number, email

## High level steps

Extract key data >

Store into database by LLM >

Extract data from database by user queries (user queries ask LLM to gen SQL to extract)

## Detailed steps

Define required data, i.e. agent name, customer email, customer order, customer phone, customer sentiment, for extraction by data class method >

Execute data class schema >

Define a database with storage location, table name, required columns, i.e. required columns contain required data >

Initialize database >

Define a update knowledge tool to update new required data in every dialogue and insert back to database >

Define a SQL execution tool to execute SQL to extract data from database >

Load a dataset >

Load a row of data for extraction and SQL checking >

Task 1 - Set up a raven prompt by importing inspect approach >

Print raven call to see extracted data >

Execute raven call to insert data back to database >

Task 2 - Set up a raven prompt by importing inspect approach >

Print raven call to see SQL code >

Evaluate raven call to run SQL code to extract the data from database >

\*Can re-initialize database and run the raven call together

Dataset: [https://huggingface.co/datasets/SantiagoPG/customer\\_service\\_chatbot](https://huggingface.co/datasets/SantiagoPG/customer_service_chatbot)