TEAM 80

TM12.0

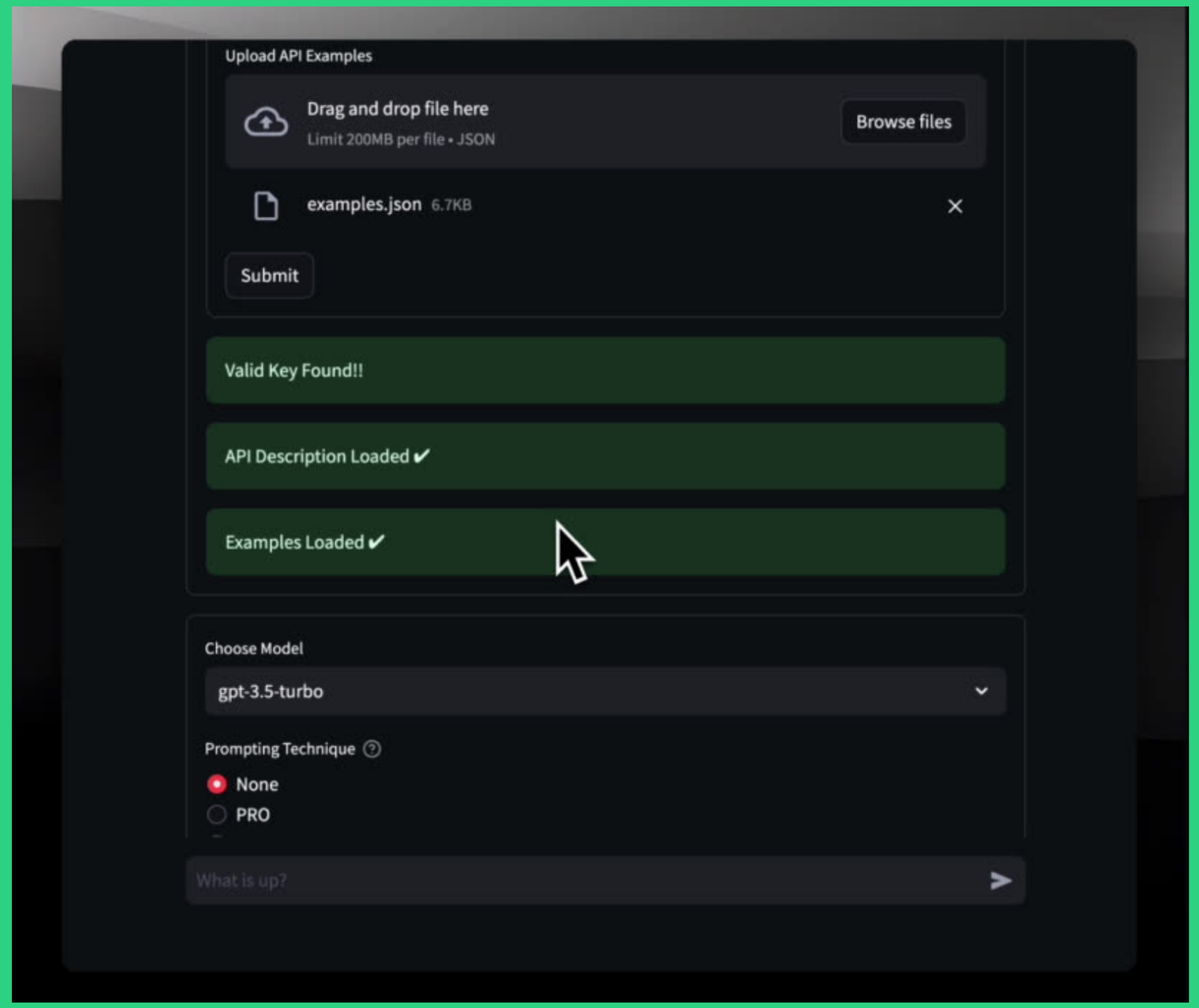
DEVREV

AI AGENT 007

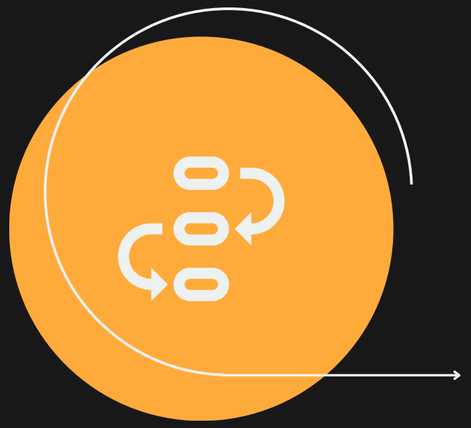
TOOLING UP FOR SUCCESS

DEMO

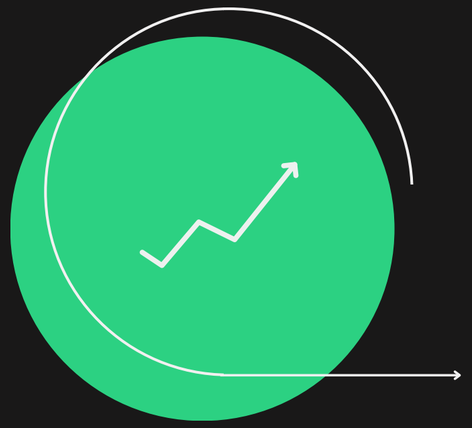
TM12.0


Upload API Examples
Drag and drop file here
Limit 200MB per file • JSON
Browse files
examples.json 6.7KB
Submit
Valid Key Found!!
API Description Loaded ✔
Examples Loaded ✔
Choose Model
gpt-3.5-turbo
Prompting Technique
None
PRO
What is up?

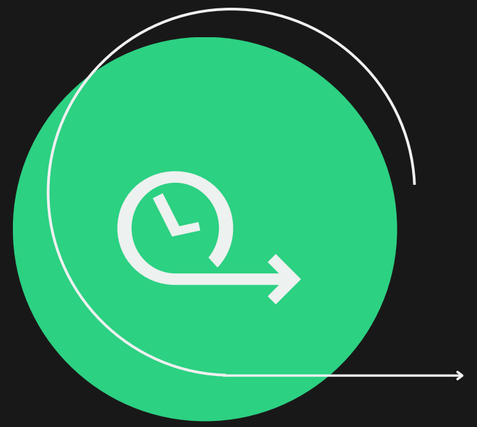
TM12.0



METHODOLOGY



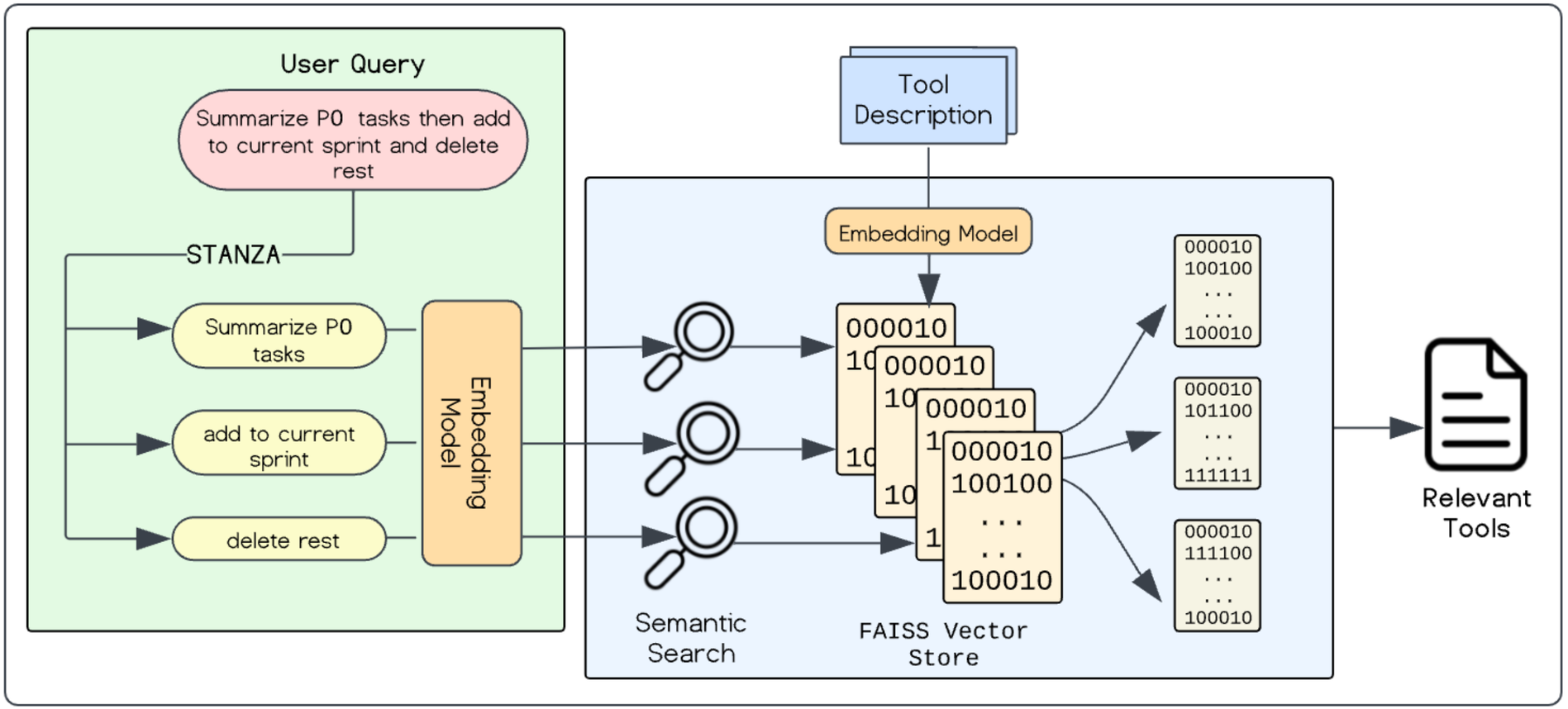
PERFORMANCE



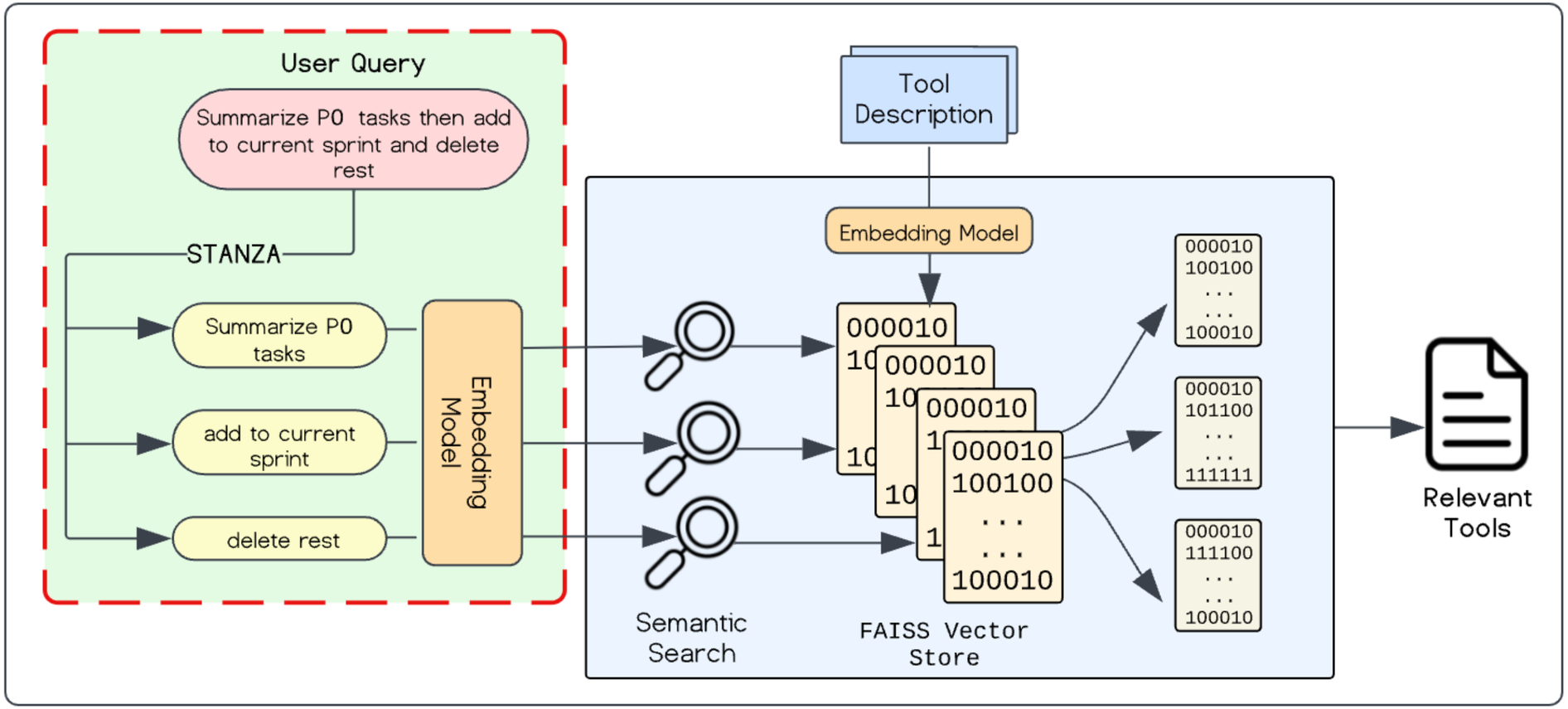
FUTURE SCOPE

METHODOLOGY

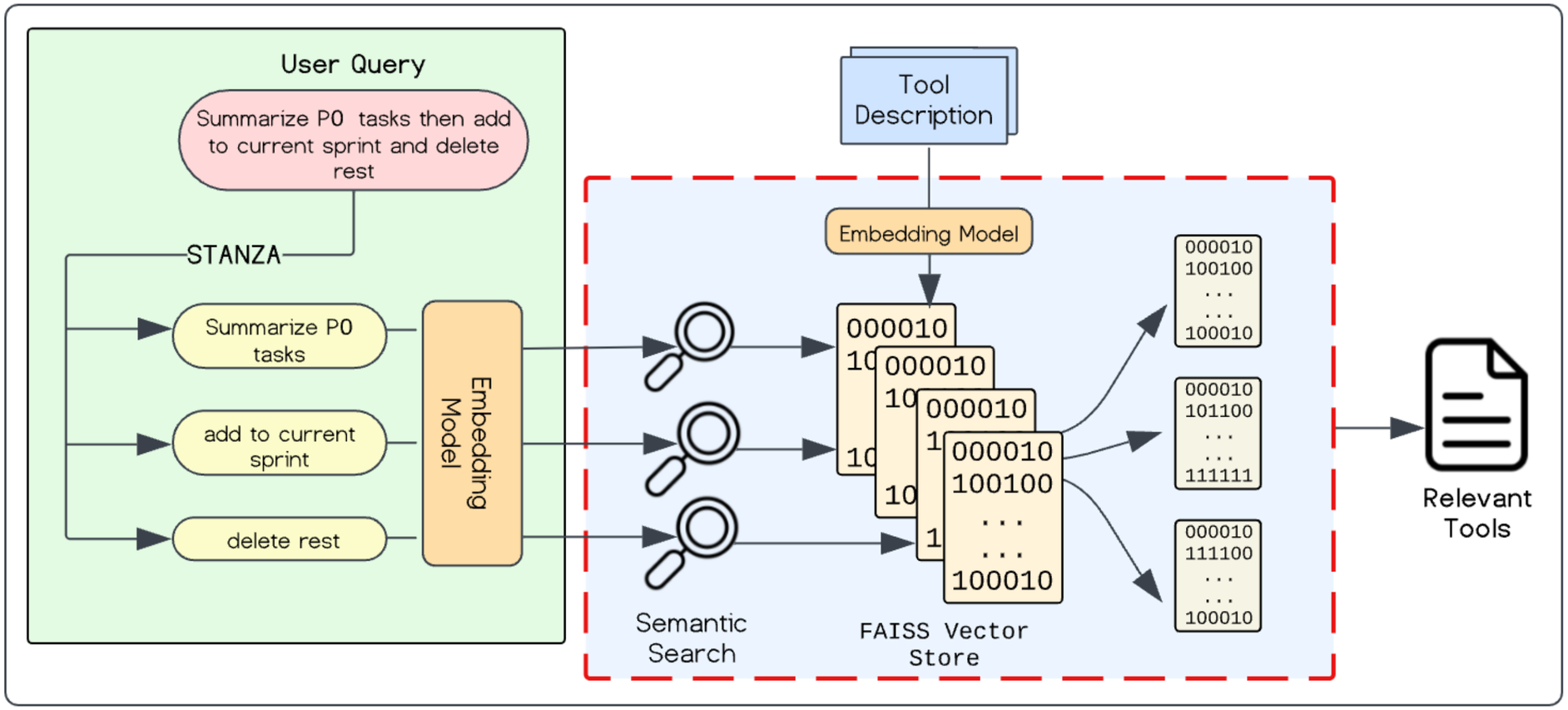
TOOL RETRIEVAL



TOOL RETRIEVAL



TOOL RETRIEVAL



TM12.0

ANSWER GENERATION

LLM SELECTION

- Chosen Model : **gpt-3.5-turbo**
- Offers best balance between accuracy and cost-efficiency

PROMPTING

- Prompts modulate model responses
- Our Method : **PIRO** Prompting
- Optimzed for **gpt-3.5-turbo**

PIRO PROMPTING

P

PLANNING

Given the list of tools, we ask the model to create an ordered skeleton of tools it needs to solve the user query.

I

IMPROVEMENT

Provided the relevant tools and examples demonstrating their usage, we ask the model to improve its answer and rectify any argument errors

R

REFLECTION

For GPT3.5, the model reflects on its answer based on the given information about tools and their examples, to fix the solution, if required.

O

OPTIMIZATION

Finally, the model optimizes its solution, and removes any redundant API calls, and unnecessary arguments.

TM12.0

PIRO PROMPTING

P

PLANNING

Given the list of tools, we ask the model to create an ordered skeleton of tools it needs to solve the user query.

I

IMPROVEMENT

Provided the relevant tools and examples demonstrating their usage, we ask the model to improve its answer and rectify any argument errors

R

REFLECTION

For GPT3.5, the model reflects on its answer based on the given information about tools and their examples, to fix the solution, if required.

O

OPTIMIZATION

Finally, the model optimizes its solution, and removes any redundant API calls, and unnecessary arguments.

TM12.0

PIRO PROMPTING

P

PLANNING

Given the list of tools, we ask the model to create an ordered skeleton of tools it needs to solve the user query.

I

IMPROVEMENT

Provided the relevant tools and examples demonstrating their usage, we ask the model to improve its answer and rectify any argument errors

R

REFLECTION

For GPT3.5, the model reflects on its answer based on the given information about tools and their examples, to fix the solution, if required.

O

OPTIMIZATION

Finally, the model optimizes its solution, and removes any redundant API calls, and unnecessary arguments.

TM12.0

PIRO PROMPTING

P

PLANNING

Given the list of tools, we ask the model to create an ordered skeleton of tools it needs to solve the user query.

I

IMPROVEMENT

Provided the relevant tools and examples demonstrating their usage, we ask the model to improve its answer and rectify any argument errors

R

REFLECTION

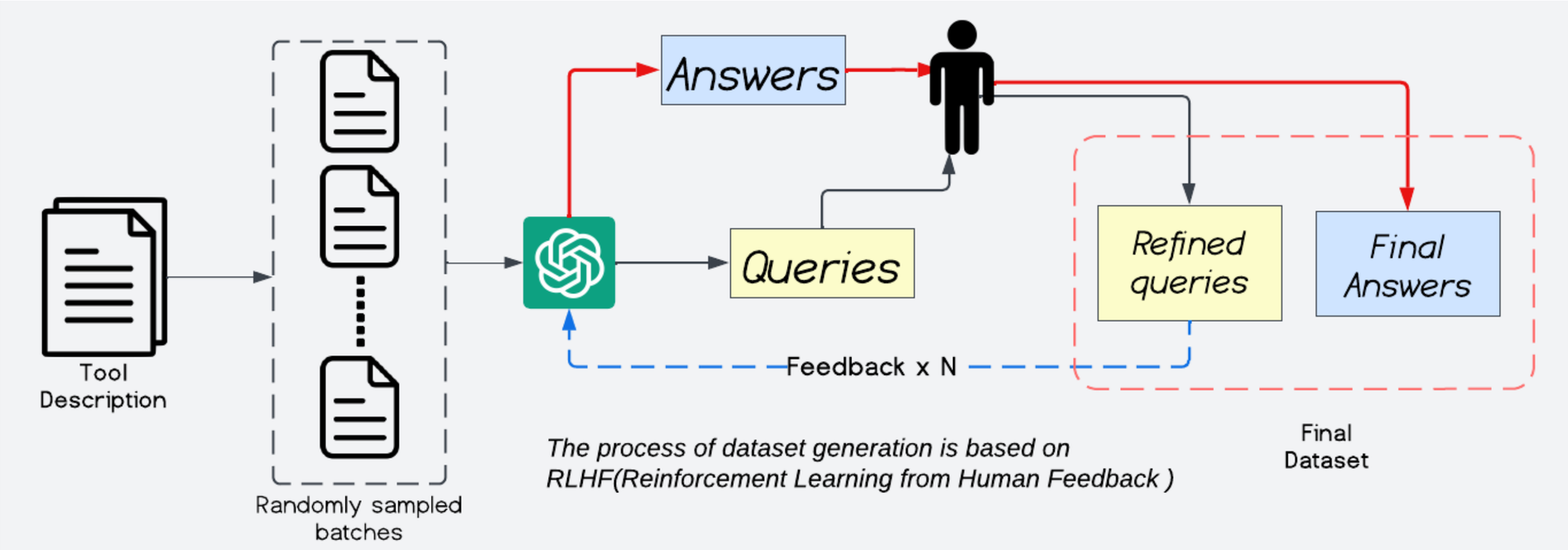
For GPT3.5, the model reflects on its answer based on the given information about tools and their examples, to fix the solution, if required.

O

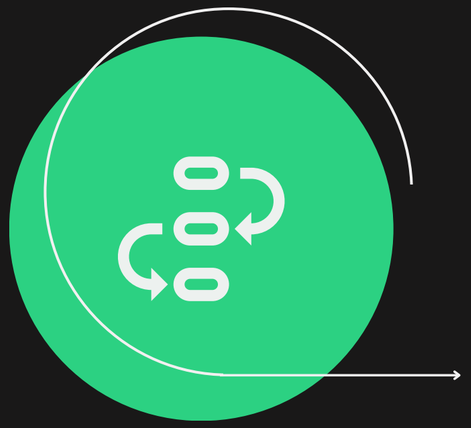
OPTIMIZATION

Finally, the model optimizes its solution, and removes any redundant API calls, and unnecessary arguments.

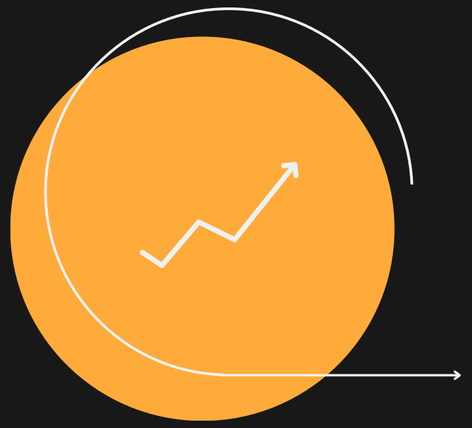
DATASET GENERATION



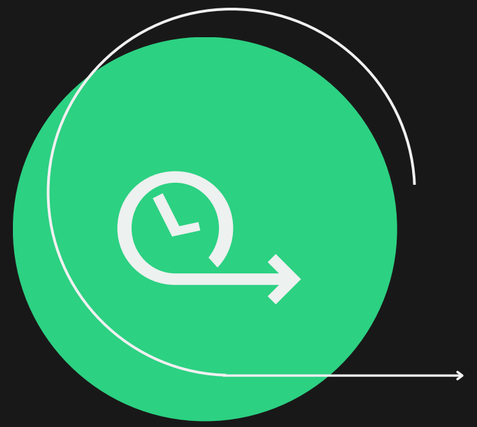
TM12.0



METHODOLOGY

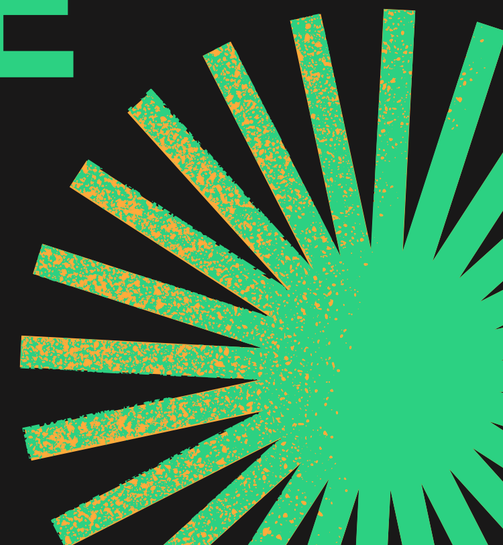


PIPELINE
PERFORMANCE



FUTURE
SCOPE

PIPELINE PERFORMANCE



TM12.0

OUR PERFORMANCE

AVERAGE INFERENCE TIME

GPT 4 (with PRO)	30s
GPT 4 Turbo (with PRO)	39s
Human Performance	76s
GPT 3.5 Turbo(with PIRO)	**19s**

ACCURACY- D1* DATASET

GPT 4 Turbo	65%
Human Performance	80%
GPT 3.5 Turbo	**65%**

ACCURACY- D2* DATASET

GPT 4 Turbo	26%
Human Performance	46%
GPT 3.5 Turbo	**24%**

* D1 - Smaller DevRev Dataset; D2 - Larger DevRev Dataset

INFERENCE COST

Model Name	Num Tokens	Cost/Query
GPT 4	5000	$ 0.165
GPT 4 Turbo	5000	$ 0.055
GPT 3.5 Turbo	6500	$ 0.007

HOW WE GOT HERE

Baseline	GPT 3.5 Turbo	25%

HOW WE GOT HERE

Baseline	GPT 3.5 Turbo	25%
PRO prompting	GPT 3.5 Turbo	40%

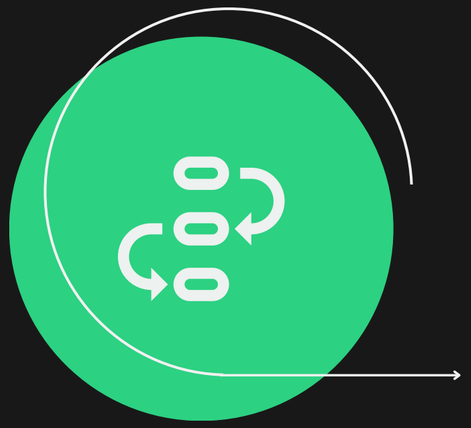
HOW WE GOT HERE

Baseline	GPT 3.5 Turbo	25%
PRO prompting	GPT 3.5 Turbo	40%
PIRO prompting	GPT 3.5 Turbo	55%

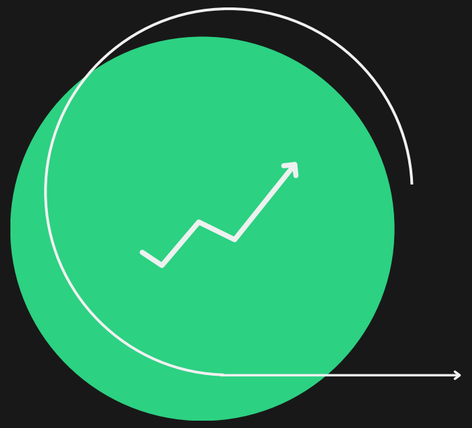
HOW WE GOT HERE

Baseline	GPT 3.5 Turbo	25%
PRO prompting	GPT 3.5 Turbo	40%
PIRO prompting	GPT 3.5 Turbo	55%
PIRO prompting+Stanza	GPT 3.5 Turbo	65%

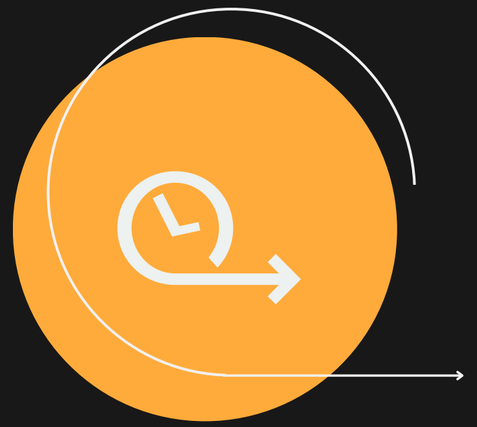
TM12.0



METHODOLOGY

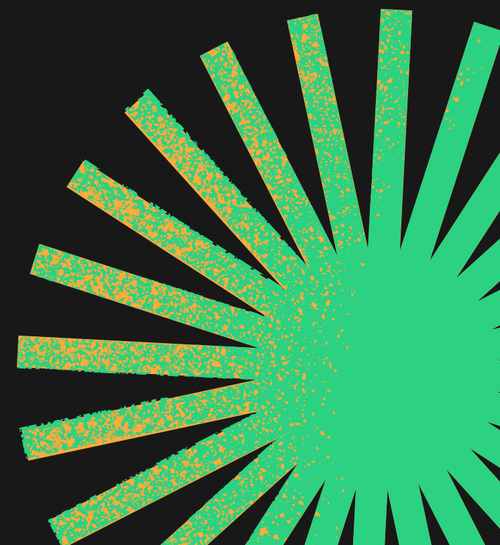


PIPELINE
PERFORMANCE



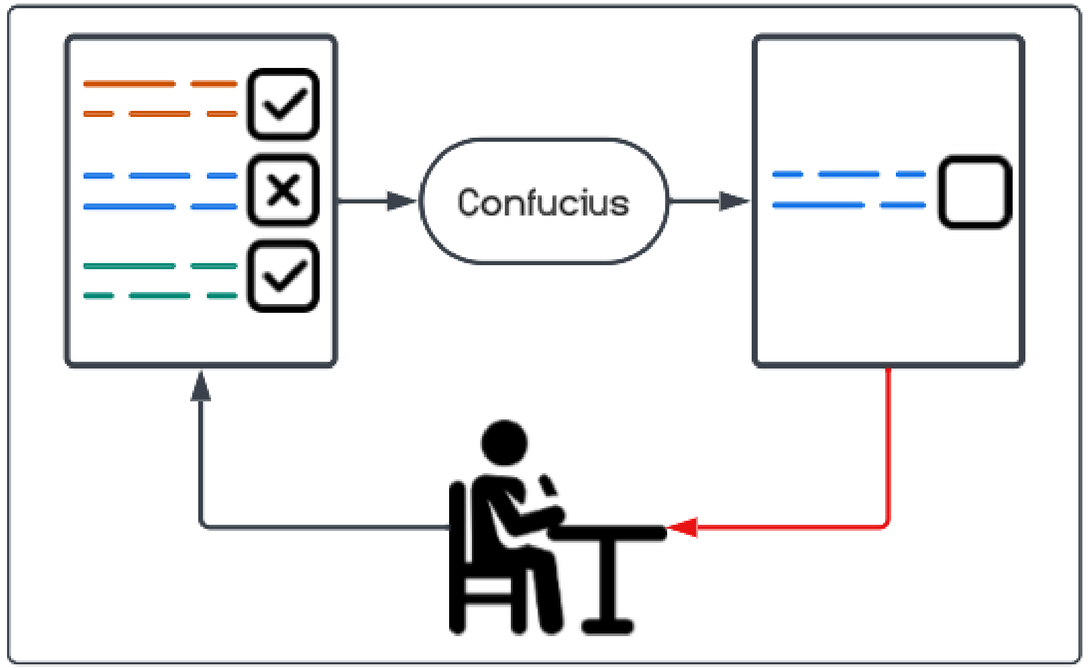
FUTURE
SCOPE

FUTURE SCOPE



TM12.0

CONFUCIUS

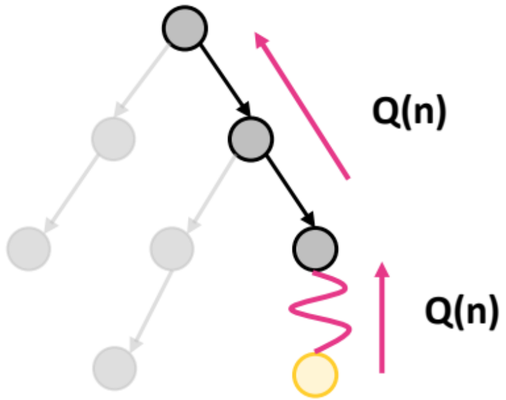


An overview of Confucius

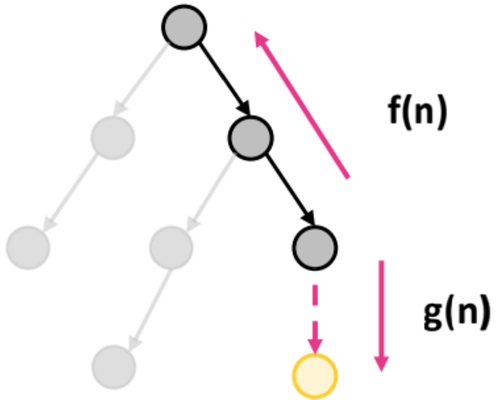
- Confucius is a tool learning framework for LLMs
- Unlike existing self-instruction methods, Confucius takes the complexity of the tool into account during learning
- It emphasizes the tools that the model struggles with to make new test sets
- The model faces more questions based on the hardest tools

TM12.0

TOOLCHAIN*



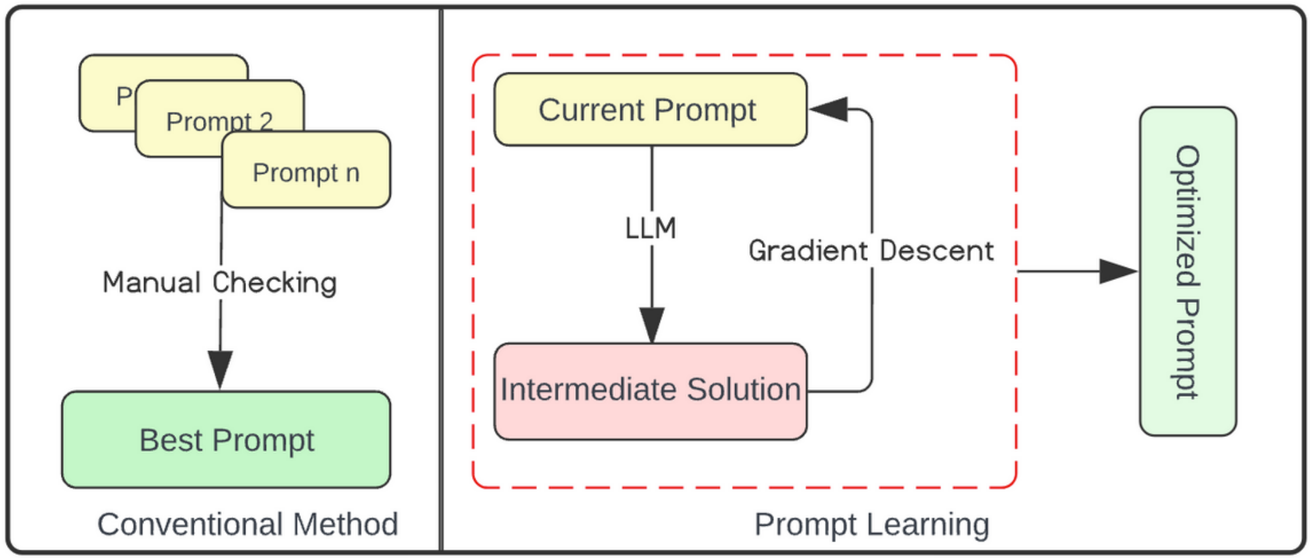
Monte Carlo Tree Search



ToolChain*

- The available search space grows exponentially with the number of APIs.
- Toolchain* aims to make this search efficient.
- It uses the expected future cost to traverse trees.
- It is faster than popular traversal algorithms like DFS and MCTS.
- It also reduces snowball effects where one small error can render the entire solution incorrect.

PROMPT LEARNING



A comparison between prompt learning and the conventional method of prompt tuning

- Prompt learning arrives at the optimal prompt using gradient descent.
- Model weights are frozen, and only prompt parameters are learnt.
- Prompt parameters are learnt through likelihood maximization.
- This is cheaper and more stable than fine tuning.

THANK YOU!

Questions are Welcome

TEAM 80