Bedrock Claude Night2!

Agentは楽しいぞ

tubone24
(@meitante1conan)

自己紹介

Yu Otsubo (tubone24)

特徴

@meitante1conan

LLM歴3ヶ月

お家DXやっているよ

それは突然やってくる!!!

いい感じの社内向け

AI Product

作ってよ

いい感じ…？

いい感じのAI Productとは

- ユーザーの課題を細かいタスクに分解し
- 社内のアセットやWeb上の情報を用い
- LLMと壁打ちをしながら
- 解決に導くことのできる製品

それって話題のAgentじゃん?

※ここでいうAgentとはReAct logicで動くLLMアプリケーションをLangChainで構築したものを指します

Bedrock Claude Night2!

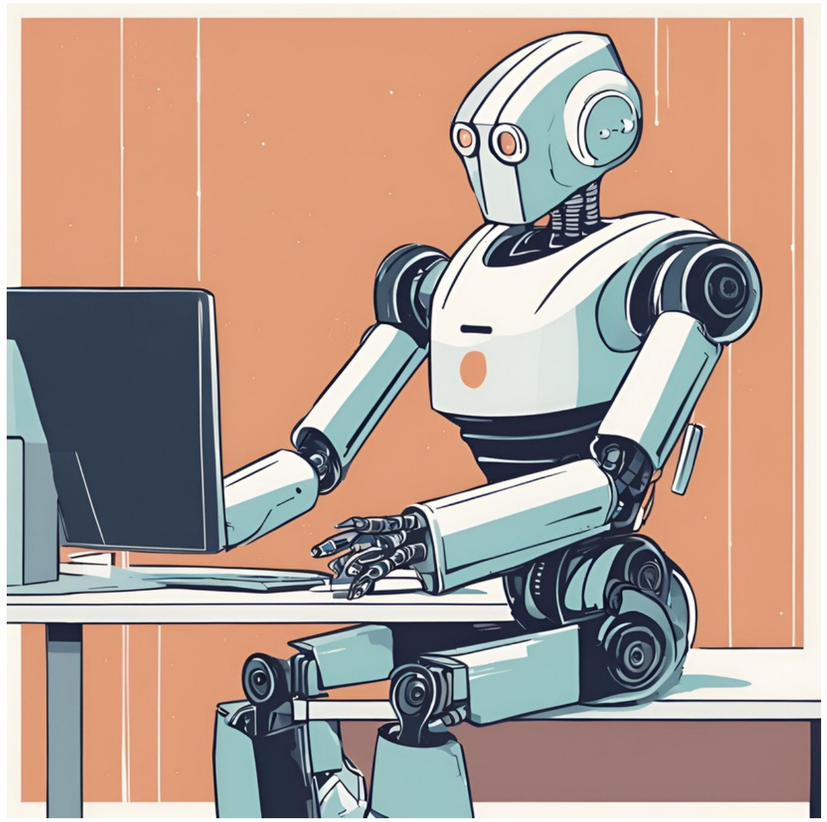
Agentは楽しいぞ

tubone24
(@meitante1conan)

Autonomous Agent(自律エージェント)とは

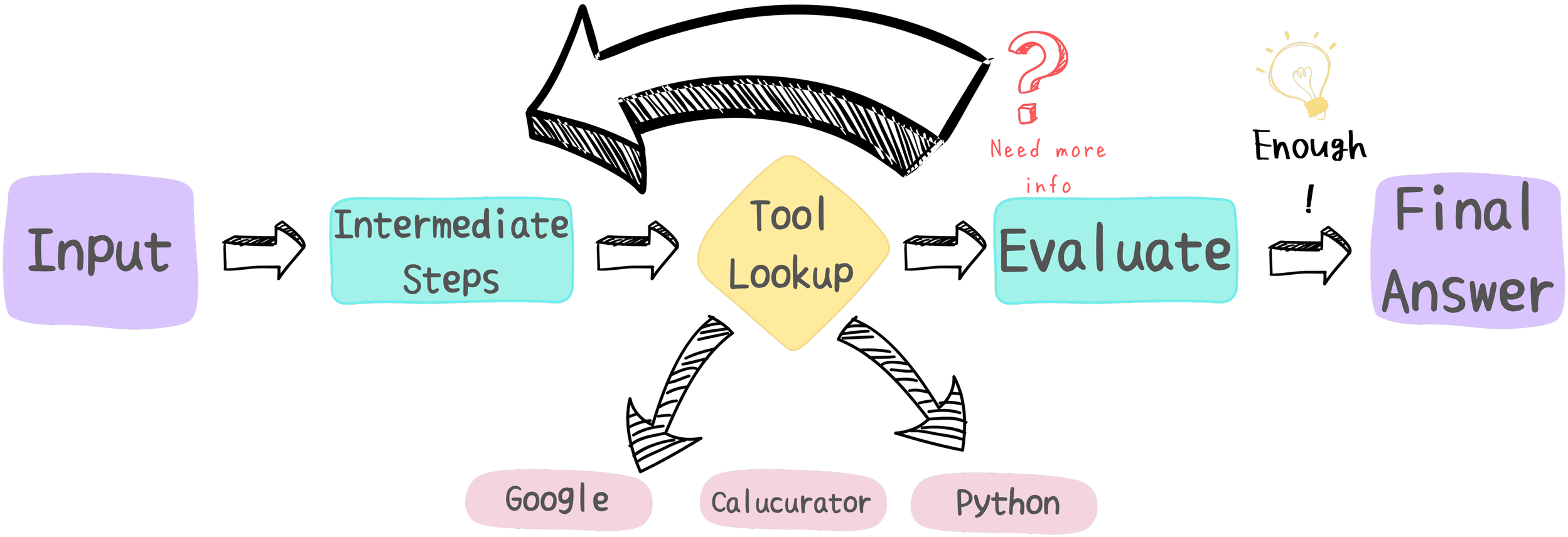
自律エージェント（Autonomous Agent）とは、何らかの環境におかれたシステムであり、その環境を感知し、自身の内的方針に従って行動する存在

Franklin, Stan and Graesser, Art (1997)
"Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents,"



いい感じに考え、
行動するやつ

ReAct-based LangChain Agents ?



でもでも…?

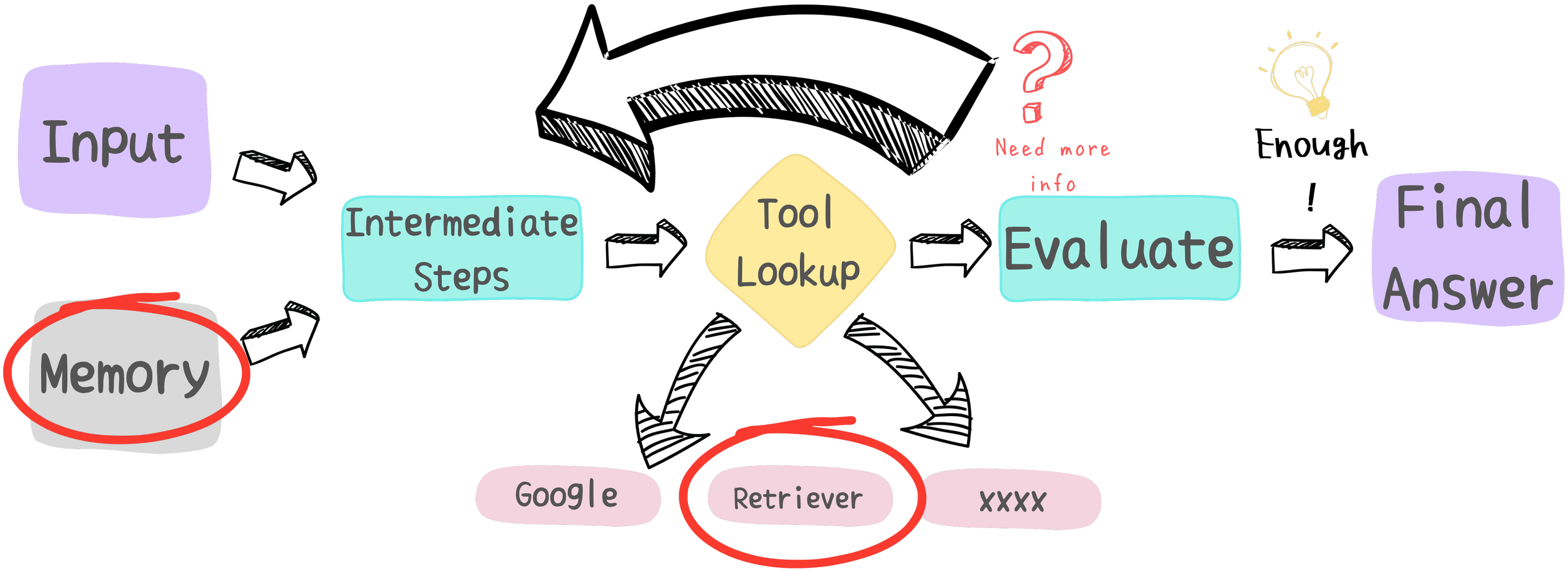
よくあるAgentとのギャップ

だいたいAgentっていうとこんなのでてくるけど
本当にほしいのはこんな感じ...

よくあるAgent	本当に欲しいAgent
LLMが問題の解決までの流れを考え、 Web検索やPythonコード実行(計算)をし 最終回答を返す	LLMが問題の解決までの流れを考え、 Web検索などのツールや 社内アセットなどを利用し、 LLMと壁打ちをしながら、 最終回答を返す

RetrieverとかMemoryとかあってこそのAgentである

ReAct-based LangChain Agents! ✨



では作ってみましょう..!!

簡単に作れます

でも課題あり

Web search(Tool)

Google検索をツールとして組み込む

Metadata Results

Run query through GoogleSearch and return snippet, title, and link metadata.

- Snippet: The description of the result.
- Title: The title of the result.
- Link: The link to the result.

```
search = GoogleSearchAPIWrapper()


def top5_results(query):
    return search.results(query, 5)


tool = Tool(
    name="Google Search Snippets",
    description="Search Google for recent results.",
    func=top5_results,
)
```

https://python.langchain.com/v0.2/docs/integrations/tools/google_search/

組み込めますが課題あり...!!!

RetrieverをAgentのツールとして組み込む

Retriever Tool

Now we need to create a tool for our retriever. The main things we need to pass in are a name for the retriever as well as a description. These will both be used by the language model, so they should be informative.

```python
from langchain.tools.retriever import create_retriever_tool

tool = create_retriever_tool(
    retriever,
    "search_state_of_union",
    "Searches and returns excerpts from the 2022 State of the Union.",
)
tools = [tool]
```

API Reference:

- create_retriever_tool

https://python.langchain.com/v0.1/docs/use_cases/question_answering/conversational_retrieval_agents/#retriever-tool

Knowledge Bases for Amazon Bedrock

Knowledge BasesをAgentのツールとして組み込む

Using the Knowledge Bases Retriever

```
%pip install --upgrade --quiet  boto3
```

```
from langchain_community.retrievers import AmazonKnowledgeBasesRetriever

retriever = AmazonKnowledgeBasesRetriever(
    knowledge_base_id="PUIJP4EQUA",
    retrieval_config={"vectorSearchConfiguration": {"numberOfResults": 4}},
)
```

API Reference: AmazonKnowledgeBasesRetriever

組み込めますが課題あり...!!!

LangChainのConversationBufferMemoryが普通に使える

Notice the usage of the `chat_history` variable in the `PromptTemplate`, which matches up with the dynamic key name in the `ConversationBufferMemory`.

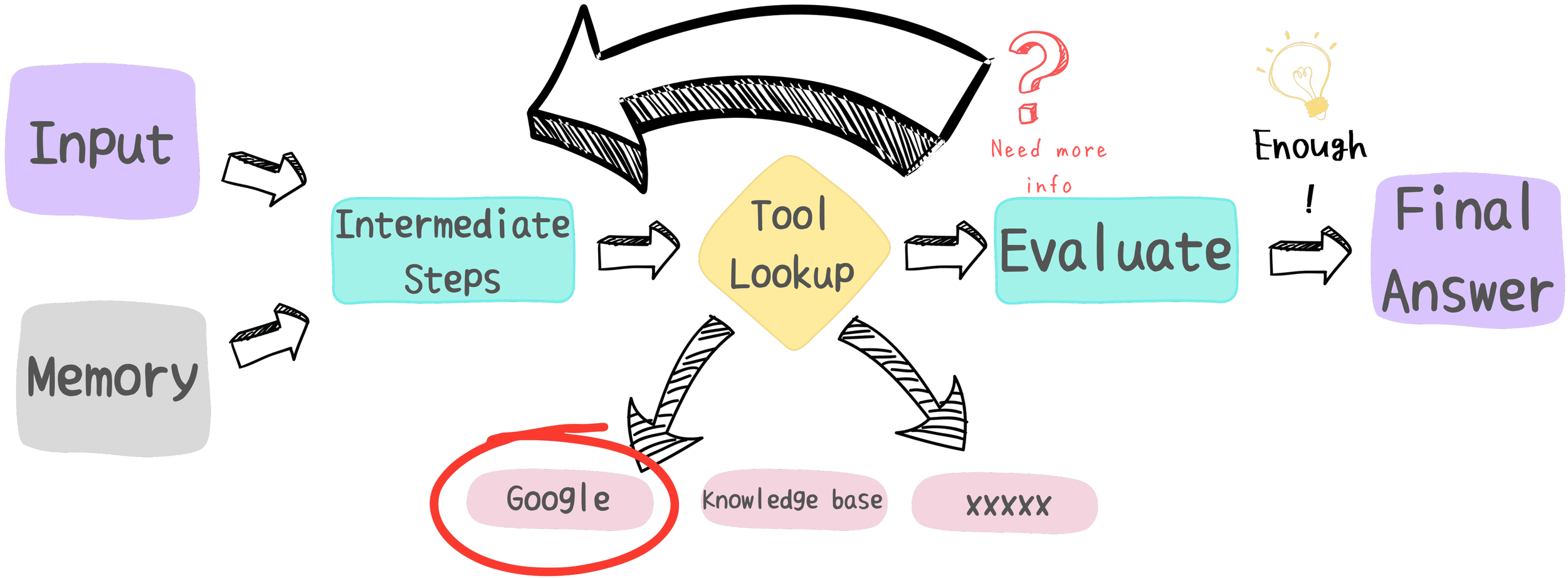
```
from langchain import hub
from langchain.agents import AgentExecutor, create_react_agent
from langchain.memory import ChatMessageHistory

prompt = hub.pull("hwchase17/react")

memory = ChatMessageHistory(session_id="test-session")
```

API Reference:

- AgentExecutor
- create_react_agent
- ChatMessageHistory

組み込めますが課題あり...!!!

課題1: Google検索結果で参照元を適切に表示しない



課題1: Google検索結果で参照元を適切に表示しない

実行した結果は情報が文字列の羅列で
取得される
参照元を適切に扱えない

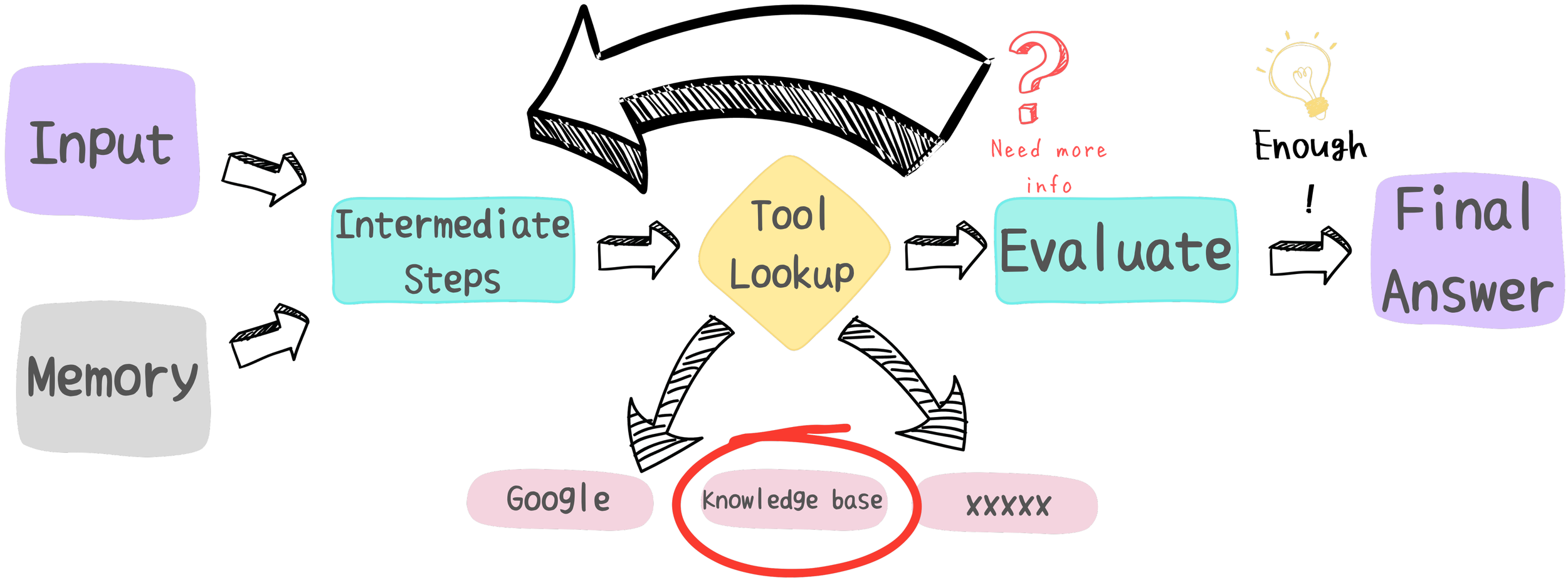
Input

"企業名 業界 概要"

Output

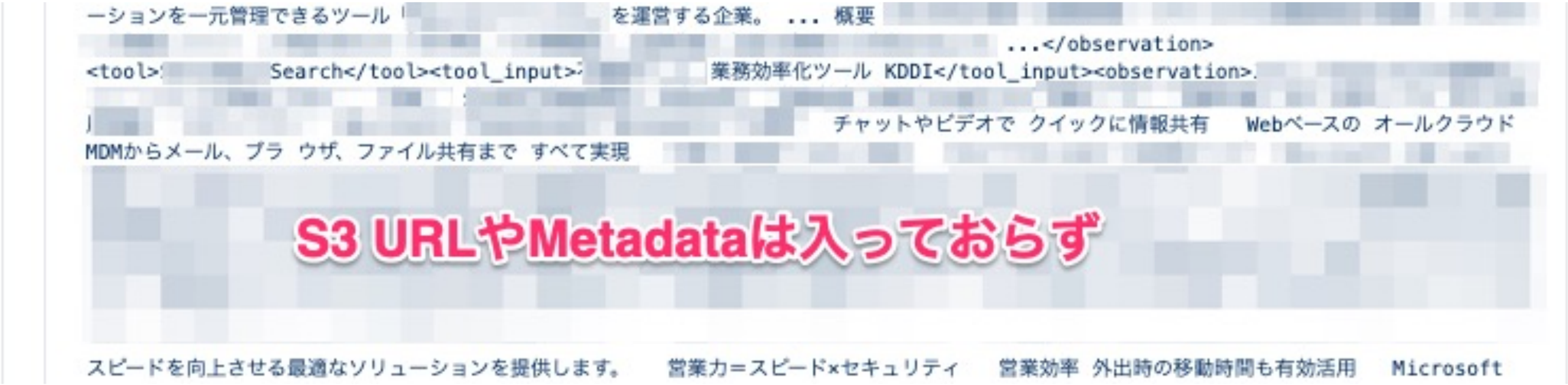
"企業概覧. 本公司成立於1972 年，最初命名為System Analysis Program Development (Systemanalyse Programmentwicklung)，後來縮寫為SAP。從那時 ... 全球知名雲端服務資料中心，使用Moxa 方案建立資料中心的機電基礎網路設施。 前十大. 油氣領導廠商. 全球前十大業界領導者中，有三名使用Moxa IIoT ... 1,202名（2023年2月末日時点、パートタイマー・アルバイトを含みません）※. 主要取引銀行，三井住友銀行. ※グループ合算の数値です。 施設一覧. 施設名，住所，電話番号 ... 会社概要. 商号. 株式会社ハンズ. Hands Inc. 事業内容. 住まいと住生活・手づくり関連の製品・道具・工具・素材・部品の総合専門小売業. 所在地（本社）. 〒160-0022 東京 ... 会社概要. 社名: シャープ株式会社（Sharp Corporation）; 本社所在地: 〒590-8522 大阪府堺市堺区 ... 事業所ファイル. 大手企業、金融機関、一部官公庁の国内の事務所を収録したサポートファイルです。 商号、事業所名、 ... Apr 1, 2024 ... 44,000百万円; 従業員数: 4,710名（グループ連結：14,487名）※2023年12月31日現在; 代表取締役社長執行役員 兼 最高経営責任者: 森 孝廣; 本社: OKI虎ノ門 ... 会社概要 · 会社名: 株式会社クレハ（KUREHA CORPORATION）· 所在地: 〒103-8552 東京都中央区日本橋浜町3-3-2 · 創立: 1944年6月21日 · 資本金: 181億6,900万円 · 売上高 ... 株式会社セブン&アイ・ホールディングス（英文名 Seven & i Holdings Co., Ltd.） 住所, 〒102-8452 東京都千代田区二番町8番地8. 電話番号, 03-6238-3000（ ... 企業理念 · 事業所案内 · 電子公告 · CSR活動 · NCAヒストリー · リンク. 会社概要. 会社名, 日本貨物航空株式会社. Nippon Cargo Airlines Co., Ltd.（略称NCA）. 代表 ..."

課題2: Knowledgebaseの参照元ファイルを表示しない

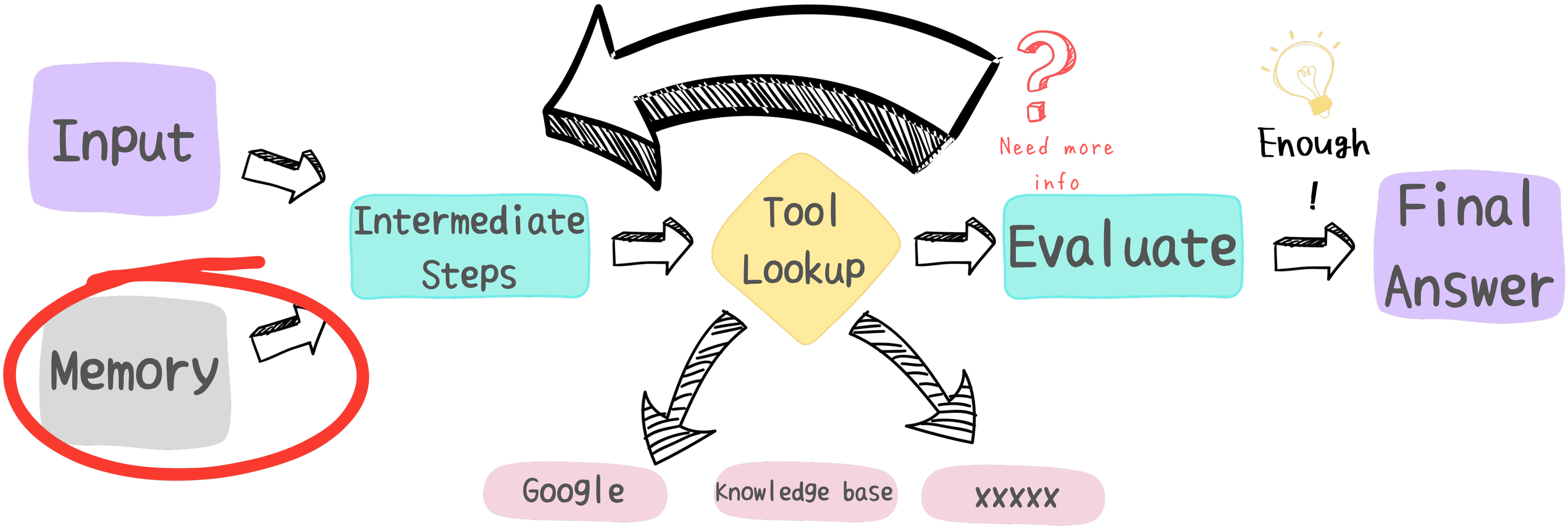


課題2: Knowledgebaseの参照元ファイルを表示しない

LangChainのAmazonKnowledgeBasesRetrieverを使うと、metadata情報がagent scratchpadから取れない



課題3: Agentのやり取りが進むとMemoryが微妙な品質になる



課題3: Agentのやり取りが進むとMemoryが微妙な品質になる

Memoryの保持の仕方がOpenAI形式になっているため、会話を続けると変なことになる

Input

```
{
  input: "どこの天気？"
  chat_history: "Human: 1週間の天気予報を教えて
  AI:

  ご質問ありがとうございます。1週間の天気予報についてお伝えします。

  日本気象協会の公式サイトtenki.jp[^1]によると、向こう1週間の全国の天気は以下の通りです。

  - 6月11日(日)：晴れのち雲多く所により雷雨
  - 6月12日(月)：晴れ時々曇り
  - 6月13日(火)：晴れ時々曇り
  - 6月14日(水)：晴れ時々曇り
  - 6月15日(木)：晴れ時々曇り
  - 6月16日(金)：晴れ時々くもり
  - 6月17日(土)：晴れのち曇り

  最高気温は25〜30度前後、最低気温は20度前後で推移する見込みです。雨の降る確率は11日以外は低めです。

  ただし地域によって差があるので、より詳細な予報は以下のリンクからご自身の地域を選んでご確認ください。

  - [2週間天気（旧:10日間天気） - 日本気象協会 tenki.jp](https://tenki.jp/week/)

  基本的に晴れる日が多い1週間になりそうですが、11日は午後から天気が崩れる所があるようなので、外出の際は折り畳み傘などの雨具をお持ちになることをおすすめします。

  参考リンク:
  [^1]: [日本気象協会 tenki.jp【公式】 / 天気・地震・台風](https://tenki.jp/)

  "
}
```

Output

!?

```
{
  log: "<tool>human</tool><tool_input>
  ビッグローブ株式会社についての理解は深まりましたが、以
```

どうする…?

ドキュメント見る

でも、Claude3 Opus用にカスタマイズすることで解決できる

公式Docsに立ち返り、Opusが理解しやすいようにXMLっぽくする

How to use XML tags

You can use XML tags to structure and delineate parts of your prompt from one another, such as separating instructions from content, or examples from instructions.

Role	Content
User	Please analyze this document and write a detailed summmary memo according to the instructions below, following the format given in the example: <document> {{DOCUMENT}} </document> <instructions> {{DETAILED_INSTRUCTIONS}} </instructions> <example> {{EXAMPLE}} </example>

https://docs.anthropic.com/ja/docs/use-xml-tags#xml

なければ実装する!!!

と言ってもちょっとした文字列の整形だけです..!

Google searchの結果をXMLタグで整形

GoogleSearchの結果をゴリゴリXMLで整形する

```
20  # Web検索のツール
21  def web_search(query: str, search_engine_type="duckduckgo"):
22      if search_engine_type == "google":
23          engine = GoogleSearchAPIWrapper()
24      elif search_engine_type == "duckduckgo":
25          engine = DuckDuckGoAPIWrapper()
26      else:
27          raise ValueError("search_engine must be 'google' or 'duckduckgo'")
28      xml_result = "<results>"
29      for item in engine.results(query, 10):
30          snippet = item["snippet"].replace("\xa0", "")
31          xml_result += "<result>"
32          xml_result += f'<title>{item["title"]}</title>'
33          xml_result += f'<url>{item["link"]}</url>'
34          xml_result += f"<snippet>{snippet}</snippet>"
35          xml_result += "</result>"
36      xml_result += "</results>"
37      return xml_result
```

boto3でKBを叩いて結果をXMLタグで整形

LangChainのライブラリを使わずboto3でゴリゴリ実装

```
28  def retrieve(query: str):
29      bedrock_agent_runtime_client = boto3.client(
30          "bedrock-agent-runtime", region_name=[redacted]
31      )
32      response = bedrock_agent_runtime_client.retrieve(
33          knowledgeBaseId=[redacted]
34          retrievalConfiguration={
35              "vectorSearchConfiguration": {
36                  "numberOfResults": 10,
37                  "overrideSearchType": "HYBRID",
38              }
39          },
40          retrievalQuery={"text": query},
41      )
42      xml_str = "<results>\n"
43      for item in response["retrievalResults"]:
44          xml_str += "<result>\n"
45          xml_str += f"<score>{item['score']}</score>\n"
46          xml_str += f"<text>{item['content']['text']}</text>\n"
47          xml_str += f"<s3location>{item['location']['s3Location']['uri']}</s3location>\n"
48          xml_str += "</result>\n"
49      xml_str += "</results>"
50      return xml_str
```

メモリもXMLで...

繰り返しですが、XMLで...

```python
class OptimizedClaude3ConversationalBufferMemory(BaseChatMemory):
    """
    Claude 3 はXML形式のデータをうまく扱える傾向があるため、ConversationalBufferMemoryを継承してClaude3用に最適化したクラス

    Ref: https://docs.anthropic.com/claude/docs/use-xml-tags
    """

    human_prefix: str = "<Human>"
    human_suffix: str = "</Human>"
    ai_prefix: str = "<AI>"
    ai_suffix: str = "</AI>"
    system_prefix: str = "<System>"
    system_suffix: str = "</System>"
    function_prefix: str = "<Function>"
    function_suffix: str = "</Function>"
    tool_prefix: str = "<Tool>"
    tool_suffix: str = "</Tool>"

    llm: BaseLanguageModel
    memory_key: str = "history"
```

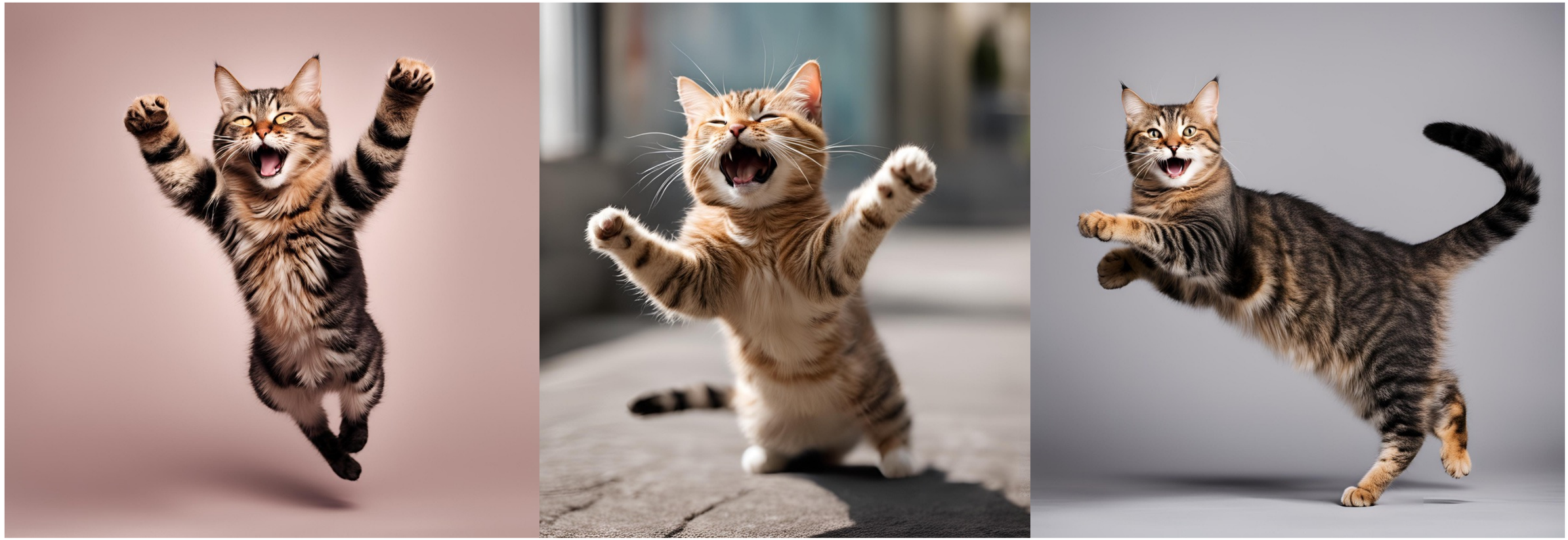
https://gist.github.com/tubone24/4ef910b152bc1017da36e7d7354e906d

ちょっとだけ開発者が歩み寄ることでOpusが輝きを放つ！

Opusの賢さがAgentの精度や
出力する日本語の質を引き上げた...!!!

社内の反応も上々
. . !!

みんにゃハッピー🐾

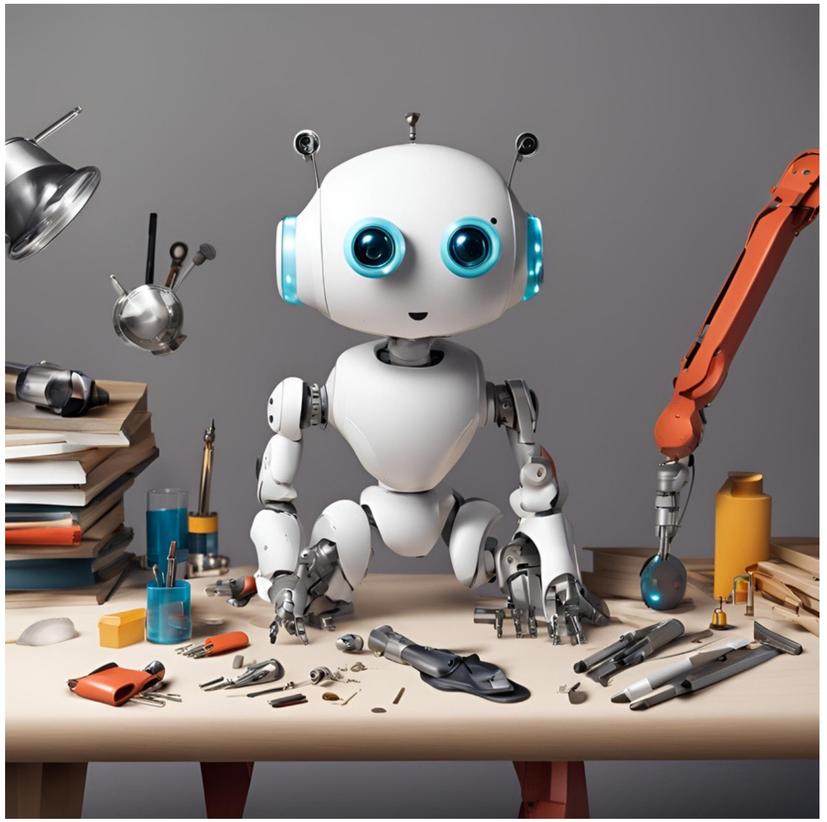


AWSのこととか本当はもっと喋りたいことがあるんですが、
それはまたの機会…!!!

まとめ

今日覚えて帰ってほしいこと

LangChainをちょっといじってOpus使えばAgentは業務でも使えるよ！



「tubone 心の声」より

ありがとうございました！