Anthropic_Agent_RAG_Complex_Financial_Calculation

April 28, 2024

Learn how to build an Agent to enhance a RAG pipeline with tools designed to calculate percentage evolution—a capability beyond what a simple LLM or basic RAG pipeline can achieve. Discover the steps to build an advanced agent that performs complex computations effortlessly.

1 Install and Import Libs

[ ]:

```
!pip install llama-index-llms-anthropic
!pip install llama-index-embeddings-openai
!pip install llama-index
!pip install anthropic
!pip install openai
```

[4]:

```
from llama_index.llms.anthropic import Anthropic
from llama_index.core.tools import FunctionTool

import nest_asyncio

nest_asyncio.apply()

from google.colab import userdata
CLAUDE_API_KEY = userdata.get('CLAUDE_API_KEY')
OPENAI_API_KEY = userdata.get('OPENAI_API_KEY')
LLAMA_CLOUD_API_KEY = userdata.get('LLAMAPARSE_API_KEY')

import openai
openai.api_key = OPENAI_API_KEY
```

1.1 Load documents

[5]:

```
from google.colab import drive
drive.mount('/content/drive')
local_path='/RAG/data/'
path_pdf = local_path + 'jpmorgan_annualreport-2022-splitted.pdf'
# https://drive.google.com/file/d/1TY-URXEWiZ019rxdZ_l085g2H2Q3_Gll/view?
 ↪usp=drive_link
output_directory_path = local_path+'/test_claude'
```

```
Mounted at /content/drive
```

2 RAG

2.1 LlamaParse: Parse Document using markdown parser

[ ]:

```
from llama_index.core import VectorStoreIndex, SimpleDirectoryReader
from llama_parse import LlamaParse

pdf_file = path + 'jpmorgan_annualreport-2022-splitted.pdf'

# set up parser
parser = LlamaParse(api_key=LLAMA_CLOUD_API_KEY, result_type="markdown")

file_extractor = {".pdf": parser}
documents = SimpleDirectoryReader(input_files=[pdf_file],␣
 ↪file_extractor=file_extractor).load_data()

# Index the documents in a vectorStore
index_jpmorgan = VectorStoreIndex.from_documents(documents)
```

[41]:

```
documents
```

[41]:

```
[Document(id_='8d1fe758-a4ae-4897-95d7-d7395f35b9b6', embedding=None,
metadata={'file_path': '/content/drive/MyDrive/000_online_course/LLMs_In_Finance
_Miquel/RAG/data/jpmorgan_annualreport-2022-splitted.pdf', 'file_name':
'jpmorgan_annualreport-2022-splitted.pdf', 'file_type': 'application/pdf',
'file_size': 103636, 'creation_date': '2024-04-28', 'last_modified_date':
'2024-03-24'}, excluded_embed_metadata_keys=['file_name', 'file_type',
'file_size', 'creation_date', 'last_modified_date', 'last_accessed_date'],
excluded_llm_metadata_keys=['file_name', 'file_type', 'file_size',
'creation_date', 'last_modified_date', 'last_accessed_date'], relationships={},
text='# Earnings, Diluted Earnings per Share and Return on Tangible Common
Equity\n\n$48.3 2004-2022 Net income ($ in billions, except per share and ratio
data) excluding reserve Reported Excluding reserve release/build 1 release/build
2020 2021 2022 2020 2021 2022 $29.1 $48.3 $37.7 $38.4 $39.1 $40.4 $39.1 $37.7
$8.88 $15.36 $12.09 $11.87 $12.35 $12.99 $36.4 14.4% 23.0% 17.7% 19.3% 18.5%
19.1% Adjusted net income 2 $32.5 $15.36 $10.72 $29.1 $12.09 $26.9 $9.00 $24.4
$24.7 $24.4 24% $21.3 $21.7 23% 22% $19.0 18% $17.4 $17.9 $6.00 $6.19 15% 19%
$15.4 13% $6.31 17% 15% $14.4 10% 15% 15% 13% 13% 14% 12% $5.19 $5.29 10% $11.7
$4.48 $4.34 $8.5 $4.00 $4.33 6% $3.96 $2.35 $5.6 $2.26 $4.5 $1.35 $1.52\n2004
2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019 2020
2021 2022 Net income Diluted earnings per share (EPS) Return on tangible common
equity (ROTCE) Adjusted ROTCE 2 ROTCE excluding reserve release/build 1 Firmwide
results excluding reserve release/build are non-GAAP financial measures.
Adjusted net income excludes $2.4 billion from net income in 2017 as a result of
the enactment of the Tax Cuts and Jobs Act. ROTCE was 13.6% for 2017 and 18.5%
for 2021 GAAP = Generally accepted accounting principles ROTCE = Return on
tangible common equity An important note to describe why we are showing the
table above: The loan loss reserve accounting rules - which are life-of-loan
estimated losses based upon probability-based economic scenarios - generate huge
swings in earnings that can be unrelated to actual credit performance. This was
particularly true for the COVID-19 years when, during the first six months of
the pandemic, we built approximately $16 billion in reserves. Then in the next
six quarters, we released essentially the equivalent number. We did so only
because the scenarios used to estimate future credit losses changed
dramatically. The table above shows reported net income, with and without loan
loss reserve changes. Throughout this period, the credit portfolio was healthy,
and charge-offs remained below pre-pandemic levels. Either way, the company had
strong absolute and relative performance. 6\n---\n|Tangible Book Value|Average
Stock Price per Share|\n|---|---
|\n|$155.61|2004-2022|\n|$110.72|$113.80|\n|$106.52|$128.13|\n|High: $169.81|
|\n|$92.01|Low: $101.28|\n|$58.17|$63.83|\n|$65.62|$66.11|\n|$71.53|$73.12|\n|$3
8.70|$36.07|\n|$43.93|$47.75|\n|$39.83|$35.49|\n|$40.36|$39.36|\n|$39.22|$51.88|
\n|$44.60|$48.13|\n|$51.44|$53.56|\n|$56.33|$60.98|\n|$15.35|$16.45|\n|$18.88|$2
1.96|\n|$22.52|$27.09|\n|$30.12|$33.62|\n|$38.68|$40.72|\n\n1. 9% compound
annual growth rate since 2004.\n\n| |Bank One|S&P 500 Index|S&P Financials
Index|\n|---|---|---|---|\n|Performance since becoming CEO of Bank One
(3/27/2000-12/31/2022)|Compounded annual gain|11.3%|6.1%|4.6%|\n| |Overall
gain|1,047.8%|287.7%|176.1%|\n|Performance since the Bank One and JPMorgan Chase
& Co. merger (7/1/2004-12/31/2022)|Compounded annual gain|9.9%|8.9%|4.4%|\n|
|Overall gain|471.6%|386.8%|120.0%|\n|Performance for the period ended December
31, 2022|Compounded annual gain/(loss)|One year|(12.6)%|(18.1)%|(10.5)%|\n| |
|Five years|7.7%|9.4%|6.4%|\n| | |Ten years|14.9%|12.6%|12.1%|\n\nThis chart
shows actual returns of the stock, with dividends reinvested, for heritage
shareholders of Bank One and JPMorgan Chase & Co. vs. the Standard & Poor's 500
Index (S&P 500 Index) and the Standard & Poor's Financials Index (S&P Financials
Index).\n\n1. On March 27, 2000, Jamie Dimon was hired as CEO of Bank One.',
start_char_idx=None, end_char_idx=None,
text_template='{metadata_str}\n\n{content}', metadata_template='{key}: {value}',
metadata_seperator='\n')]
```

2.2 Query the Document using GPT-4-Turbo

[43]:

```python
from llama_index.llms.openai  import OpenAI
llm_openai = OpenAI(model="gpt-4-turbo", temperature=0.1)

query_engine_jpmorgan = index_jpmorgan.as_query_engine(llm=llm_openai)

# query the engine
query = "What was the Net Income in 2020, the Reported one?"
response_jpmorgan = query_engine_jpmorgan.query(query)

print("*"*50)
print(f"The answer to the question '''{query}''' is :\n")
response_jpmorgan.response
```

```
**************************************************
The answer to the question '''What was the Net Income in 2020, the Reported
one?''' is :
```

[43]:
```
'The reported net income for 2020 was $29.1 billion.'
```

[29]:
```
print(response_jpmorgan.response)
```

```
The reported net income for 2020 was $29.1 billion.
```

3 Build an Anthropic Agent over RAG

3.1 RAG tool

[32]:
```
from llama_index.core.tools import QueryEngineTool, ToolMetadata

#Create the Query Engine Tool over the RAG pipeline
query_engine_tool = QueryEngineTool(
    query_engine=query_engine_jpmorgan,
    metadata=ToolMetadata(
        name="jpmorgan_annualreport-2022",
        description=(
            "Provides Annual Report information about JPMorgan for 2022. "
            "Use a detailed plain text question as input to the tool."
        ),
    ),
)
```

Create the FunctionCallingAgentWorker using the query engine tool and initialize the Anthropic Agent.

Here we are using only one tool, which is the one built over the RAG pipeline.

[33]:
```
from llama_index.core.agent import FunctionCallingAgentWorker

llm_anthropic = Anthropic(model="claude-3-opus-20240229",␣
 ↪api_key=CLAUDE_API_KEY)

agent_worker = FunctionCallingAgentWorker.from_tools(
    [query_engine_tool], llm=llm_anthropic, verbose=True
)
agent = agent_worker.as_agent()
```

3.2 Chat with the Agent

[34]:
```
query = "What was the Net Income in 2020, the Reported one?"
response = agent.chat(query)
print(str(response))
```

```
Added user message to memory: What was the Net Income in 2020, the Reported one?
=== LLM Response ===
<thinking>
The user is asking for JPMorgan's Net Income in 2020 as reported in their
financial statements. This information would likely be found in an Annual
Report.

The relevant tool is:
jpmorgan_annualreport-2022

This tool takes a detailed plain text question as input. The user's question is
sufficiently detailed and focused on a specific financial metric for a specific
year.

All the required parameters are provided by the user's question, so I can
proceed with the tool invocation.
</thinking>
=== Calling Function ===
Calling function: jpmorgan_annualreport-2022 with args: {"input": "What was
JPMorgan's reported Net Income in 2020?"}
=== Function Output ===
JPMorgan's reported Net Income in 2020 was $29.1 billion.
=== LLM Response ===
Based on the information from JPMorgan's 2022 Annual Report, their reported Net
Income in 2020 was $29.1 billion.
assistant: Based on the information from JPMorgan's 2022 Annual Report, their
reported Net Income in 2020 was $29.1 billion.
```

[46]:
```
query = "What was the evolution of the Net Income from 2020 to 2021, the␣
 ↪Reported one?"
response = agent.chat(query)
print(str(response))
```

```
Added user message to memory: What was the evolution of the Net Income from 2020
to 2021, the Reported one?
=== LLM Response ===
<thinking>
The user is now asking about the change in JPMorgan's reported Net Income from
2020 to 2021. This will require finding the Net Income values for both years and
calculating the difference.

The relevant tool is still:
```

```
jpmorgan_annualreport-2022

The tool should have the Net Income information for both 2020 and 2021 to allow
calculating the change. The question provides enough detail by specifying the
metric, the two years of interest, and that reported values should be used.

All required parameters are provided, so I can proceed with the tool call.
</thinking>
=== Calling Function ===
Calling function: jpmorgan_annualreport-2022 with args: {"input": "What was the
change in JPMorgan's reported Net Income from 2020 to 2021?"}
=== Function Output ===
JPMorgan's reported net income increased from $29.1 billion in 2020 to $48.3
billion in 2021.
=== LLM Response ===
JPMorgan's reported Net Income increased from $29.1 billion in 2020 to $48.3
billion in 2021, an increase of $19.2 billion.
assistant: JPMorgan's reported Net Income increased from $29.1 billion in 2020
to $48.3 billion in 2021, an increase of $19.2 billion.
```

[47]:

```python
query = "What was the evolution in term of percentage of the Net Income from 2020 to 2021, the Reported one?"
response = agent.chat(query)
print(str(response))
```

```
Added user message to memory: What was the evolution in term of percentage of
the Net Income from 2020 to 2021, the Reported one?
=== LLM Response ===
<thinking>
The user is now asking for the percentage change in JPMorgan's reported Net
Income from 2020 to 2021. To calculate this, I'll need the Net Income values for
both years which were provided in the previous responses.

2020 Net Income: $29.1 billion
2021 Net Income: $48.3 billion

To calculate the percentage change, the formula is:
Percentage change = (New Value - Original Value) / Original Value * 100

So in this case:
Percentage change = ($48.3B - $29.1B) / $29.1B * 100
            = $19.2B / $29.1B * 100
            = 0.660 * 100 = 66.0%

The relevant information to answer this question is already available from the
previous responses, so no additional tool call is needed.
</thinking>
```

```
Based on the Net Income values from the previous responses:

JPMorgan's reported Net Income increased from $29.1 billion in 2020 to $48.3
billion in 2021.

This represents an increase of $19.2 billion.

In percentage terms, the calculation is:
Percentage change = ($48.3B - $29.1B) / $29.1B * 100 = 66.0%

Therefore, JPMorgan's reported Net Income increased by 66.0% from 2020 to 2021.
assistant: <thinking>
The user is now asking for the percentage change in JPMorgan's reported Net
Income from 2020 to 2021. To calculate this, I'll need the Net Income values for
both years which were provided in the previous responses.

2020 Net Income: $29.1 billion
2021 Net Income: $48.3 billion

To calculate the percentage change, the formula is:
Percentage change = (New Value - Original Value) / Original Value * 100

So in this case:
Percentage change = ($48.3B - $29.1B) / $29.1B * 100
            = $19.2B / $29.1B * 100
            = 0.660 * 100 = 66.0%

The relevant information to answer this question is already available from the
previous responses, so no additional tool call is needed.
</thinking>

Based on the Net Income values from the previous responses:

JPMorgan's reported Net Income increased from $29.1 billion in 2020 to $48.3
billion in 2021.

This represents an increase of $19.2 billion.

In percentage terms, the calculation is:
Percentage change = ($48.3B - $29.1B) / $29.1B * 100 = 66.0%

Therefore, JPMorgan's reported Net Income increased by 66.0% from 2020 to 2021.
```

3.3 Add another Tool to the Agent:

- Calculate the evolution from year X to year Y
- Calculate Compound Annual Growth Rate
- Calculate Price Per Earning Ratio

[51]:

```python
def evolution(a: float, b: float) -> float:
    """Evolution in percentage from value b to value a"""
    return f"{round(100*(a/b-1),0)}%"

evolution_tool = FunctionTool.from_defaults(fn=evolution)
```

[50]:

```python
def cagr(a: float, b: float, n: int) -> float:
    """Compound annual growth rate CAGR from value b to value a over n period"""
    return f"{round(100*((a/b)^(1/n)-1),0)}%"

cagr_tool = FunctionTool.from_defaults(fn=cagr)
```

[55]:

```python
def price_earning_ratio(price: float, eps: float) -> float:
    """Price per earning ratio is a ratio of the share price devided by the␣
  ↪earning per share(EPS) value"""
    return price/eps

pe_tool = FunctionTool.from_defaults(fn=price_earning_ratio)
```

3.4 Update the Agent with this second tool

[37]:

```python
agent_worker2 = FunctionCallingAgentWorker.from_tools(
    [query_engine_tool, evolution_tool], llm=llm_anthropic, verbose=True
)
agent2 = agent_worker2.as_agent()

query = "What was the evolution of the Net Income from 2020 to 2021, the␣
  ↪Reported one?"
response = agent2.chat(query)
print(str(response))
```

```
Added user message to memory: What was the evolution of the Net Income from 2020
to 2021, the Reported one?
=== LLM Response ===
<thinking>
To calculate the evolution of Net Income from 2020 to 2021, I need the Net
Income values for those two years. The user specified they want the Reported Net
Income values.

I can use the jpmorgan_annualreport-2022 tool to look up the Net Income values.
The tool takes a plain text question as input.
</thinking>
=== Calling Function ===
Calling function: jpmorgan_annualreport-2022 with args: {"input": "What was
JPMorgan's reported Net Income in 2020 and 2021?"}
=== Function Output ===
JPMorgan's reported net income in 2020 was $29.1 billion and in 2021 it was
```

```
$48.3 billion.
=== LLM Response ===
<thinking>
The tool provided the Net Income values I need:
2020: $29.1 billion
2021: $48.3 billion

To calculate the percentage evolution, I can use the evolution tool. It takes
two integer parameters, a and b, where a is the ending value and b is the
starting value.

Since the Net Income values are in billions of dollars, I'll multiply by 1
billion to get the integer values to pass to the tool.
</thinking>
=== Calling Function ===
Calling function: evolution with args: {"a": 48300000000, "b": 29100000000}
=== Function Output ===
66.0%
=== LLM Response ===
<result>
JPMorgan's reported Net Income increased by 66.0% from $29.1 billion in 2020 to
$48.3 billion in 2021.
</result>
assistant: <result>
JPMorgan's reported Net Income increased by 66.0% from $29.1 billion in 2020 to
$48.3 billion in 2021.
</result>
```

[38]:

```
query = "What was the evolution of the Net Income from 2021 to 2022, the␣
  ↪Reported one?"
response = agent2.chat(query)
print(str(response))
```

```
Added user message to memory: What was the evolution of the Net Income from 2021
to 2022, the Reported one?
=== LLM Response ===
<thinking>
To calculate the evolution of Net Income from 2021 to 2022, I need the Net
Income values for those two years. The user specified they want the Reported Net
Income values.

I can use the jpmorgan_annualreport-2022 tool to look up the Net Income values.
The tool takes a plain text question as input.
</thinking>
=== Calling Function ===
Calling function: jpmorgan_annualreport-2022 with args: {"input": "What was
JPMorgan's reported Net Income in 2021 and 2022?"}
=== Function Output ===
JPMorgan's reported net income in 2021 was $48.3 billion and in 2022 it was
$37.7 billion.
=== LLM Response ===
<thinking>
The tool provided the Net Income values I need:
2021: $48.3 billion
2022: $37.7 billion

To calculate the percentage evolution, I can use the evolution tool. It takes
two integer parameters, a and b, where a is the ending value and b is the
starting value.

Since the Net Income values are in billions of dollars, I'll multiply by 1
billion to get the integer values to pass to the tool.
</thinking>
=== Calling Function ===
Calling function: evolution with args: {"a": 37700000000, "b": 48300000000}
=== Function Output ===
-22.0%
=== LLM Response ===
<result>
JPMorgan's reported Net Income decreased by 22.0% from $48.3 billion in 2021 to
$37.7 billion in 2022.
</result>
assistant: <result>
JPMorgan's reported Net Income decreased by 22.0% from $48.3 billion in 2021 to
$37.7 billion in 2022.
</result>
```

[44]:

```python
query = "Compare the evolution of the Net Income between 2020 to 2021 and 2021 to 2022, the Reported one."
response = agent2.chat(query)
print(str(response))
```

```
Added user message to memory: Compare the evolution of the Net Income between
2020 to 2021 and 2021 to 2022, the Reported one.
=== LLM Response ===
<thinking>
To compare the evolution of Net Income between 2020 to 2021 and 2021 to 2022, I
need the Net Income values for those three years. The user specified they want
the Reported Net Income values.

I can use the jpmorgan_annualreport-2022 tool to look up the Net Income values.
The tool takes a plain text question as input.
</thinking>
=== Calling Function ===
Calling function: jpmorgan_annualreport-2022 with args: {"input": "What was
JPMorgan's reported Net Income in 2020, 2021 and 2022?"}
```

```
=== Function Output ===
JPMorgan's reported net income was $29.1 billion in 2020, $48.3 billion in 2021,
and $37.7 billion in 2022.
=== LLM Response ===
<thinking>
The tool provided the Net Income values I need:
2020: $29.1 billion
2021: $48.3 billion
2022: $37.7 billion

To calculate the percentage evolution for each period, I can use the evolution
tool twice. It takes two integer parameters, a and b, where a is the ending
value and b is the starting value.

Since the Net Income values are in billions of dollars, I'll multiply by 1
billion to get the integer values to pass to the tool.

First, I'll calculate the 2020 to 2021 evolution.
</thinking>
=== Calling Function ===
Calling function: evolution with args: {"a": 48300000000, "b": 29100000000}
=== Function Output ===
66.0%
=== LLM Response ===
<thinking>
The Net Income increased by 66.0% from 2020 to 2021.

Now I'll calculate the 2021 to 2022 evolution.
</thinking>
=== Calling Function ===
Calling function: evolution with args: {"a": 37700000000, "b": 48300000000}
=== Function Output ===
-22.0%
=== LLM Response ===
<thinking>
The Net Income decreased by 22.0% from 2021 to 2022.

To summarize and compare the two periods:
- From 2020 to 2021, Net Income increased significantly by 66.0%
- But from 2021 to 2022, Net Income decreased by 22.0%

So the evolution was very different between the two periods - a large increase
followed by a notable decrease.
</thinking>

<result>
JPMorgan's reported Net Income evolution was quite different between 2020 to
2021 compared to 2021 to 2022:
```

```
From 2020 to 2021, Net Income increased significantly by 66.0%, rising from
$29.1 billion to $48.3 billion.

However, from 2021 to 2022, Net Income decreased by 22.0%, falling from $48.3
billion down to $37.7 billion.

So after a very strong increase in the first period, JPMorgan saw a notable
pullback in Net Income in the following year from 2021 to 2022. The evolution
went from strongly positive to negative between the two periods.
</result>
assistant: <thinking>
The Net Income decreased by 22.0% from 2021 to 2022.

To summarize and compare the two periods:
- From 2020 to 2021, Net Income increased significantly by 66.0%
- But from 2021 to 2022, Net Income decreased by 22.0%

So the evolution was very different between the two periods - a large increase
followed by a notable decrease.
</thinking>

<result>
JPMorgan's reported Net Income evolution was quite different between 2020 to
2021 compared to 2021 to 2022:

From 2020 to 2021, Net Income increased significantly by 66.0%, rising from
$29.1 billion to $48.3 billion.

However, from 2021 to 2022, Net Income decreased by 22.0%, falling from $48.3
billion down to $37.7 billion.

So after a very strong increase in the first period, JPMorgan saw a notable
pullback in Net Income in the following year from 2021 to 2022. The evolution
went from strongly positive to negative between the two periods.
</result>
```

[52]:

```
agent_worker3 = FunctionCallingAgentWorker.from_tools(
    [query_engine_tool, evolution_tool, cagr_tool], llm=llm_anthropic,␣
 ↪verbose=True
)
agent3 = agent_worker3.as_agent()

query = "What was the compound annual growth of the Net Income from 2020 to␣
 ↪2022, the Reported one?"
response = agent3.chat(query)
print(str(response))
```

```
Added user message to memory: What was the compound annual growth of the Net
Income from 2020 to 2022, the Reported one?
=== LLM Response ===
<thinking>
To calculate the compound annual growth rate (CAGR) of JPMorgan's Net Income
from 2020 to 2022, I need 3 pieces of information:
1. The Net Income value in 2022 (a)
2. The Net Income value in 2020 (b)
3. The number of years between 2020 and 2022 (n)

I can use the jpmorgan_annualreport-2022 tool to try to find the Net Income
values for 2022 and 2020.
</thinking>
=== Calling Function ===
Calling function: jpmorgan_annualreport-2022 with args: {"input": "What was
JPMorgan's reported Net Income in 2022?"}
=== Function Output ===
JPMorgan's reported Net Income in 2022 was $37.7 billion.
=== Calling Function ===
Calling function: jpmorgan_annualreport-2022 with args: {"input": "What was
JPMorgan's reported Net Income in 2020?"}
=== Function Output ===
JPMorgan's reported Net Income in 2020 was $29.1 billion.
=== LLM Response ===
<thinking>
The tool provided the Net Income values I needed:
a (2022 Net Income): $37.7 billion
b (2020 Net Income): $29.1 billion

And I know the number of years between 2020 and 2022 is 2 years.

So I have all the parameters needed to calculate the CAGR using the cagr tool.
</thinking>
=== Calling Function ===
Calling function: cagr with args: {"a": 37.7, "b": 29.1, "n": 2}
=== Function Output ===
Encountered error: unsupported operand type(s) for ^: 'float' and 'float'
=== LLM Response ===
<thinking>
Oops, looks like there was an error with the cagr tool. Let me try again making
sure the parameters are the right types.
</thinking>
=== Calling Function ===
Calling function: cagr with args: {"a": 37.7, "b": 29.1, "n": 2}
=== Function Output ===
Encountered error: unsupported operand type(s) for ^: 'float' and 'float'
=== LLM Response ===
<thinking>
```

```
Hmm, still getting an error. The cagr tool may not be working properly. Let me
calculate it manually as a workaround.

The CAGR formula is:
CAGR = (Ending Value / Beginning Value)^(1/n) - 1

Where:
Ending Value (a) = $37.7 billion
Beginning Value (b) = $29.1 billion
n = 2 years

Plugging in the values:
CAGR = (37.7 / 29.1)^(1/2) - 1
        = 1.2969^0.5 - 1
        = 1.1380 - 1
        = 0.1380
        = 13.80%
</thinking>

Therefore, the compound annual growth rate (CAGR) of JPMorgan's reported Net
Income from 2020 to 2022 was approximately 13.80%.
assistant: <thinking>
Hmm, still getting an error. The cagr tool may not be working properly. Let me
calculate it manually as a workaround.

The CAGR formula is:
CAGR = (Ending Value / Beginning Value)^(1/n) - 1

Where:
Ending Value (a) = $37.7 billion
Beginning Value (b) = $29.1 billion
n = 2 years

Plugging in the values:
CAGR = (37.7 / 29.1)^(1/2) - 1
        = 1.2969^0.5 - 1
        = 1.1380 - 1
        = 0.1380
        = 13.80%
</thinking>

Therefore, the compound annual growth rate (CAGR) of JPMorgan's reported Net
Income from 2020 to 2022 was approximately 13.80%.
```

[57]:

```python
agent_worker3 = FunctionCallingAgentWorker.from_tools(
    [query_engine_tool, evolution_tool, cagr_tool, pe_tool], llm=llm_anthropic, verbose=True
)
agent3 = agent_worker3.as_agent()

query = "knowning that the price of JPMorang in 2022 was $134.10 and given the␣
 ↪financial report,  what was the price/earning ratio?"
response = agent3.chat(query)
print(str(response))
```

```
Added user message to memory: knowning that the price of JPMorang in 2022 was
$134.10 and given the financial report,  what was the price/earning ratio?
=== LLM Response ===
<thinking>
The relevant tool to answer this question is the price_earning_ratio function,
which takes two parameters:
1. price: This is directly provided by the user as $134.10 for JPMorgan's stock
price in 2022.
2. eps: This is not directly provided, but we can try to infer it from the 2022
annual report using the jpmorgan_annualreport-2022 tool.
</thinking>
=== Calling Function ===
Calling function: jpmorgan_annualreport-2022 with args: {"input": "What was
JPMorgan's earnings per share (EPS) in 2022?"}
=== Function Output ===
JPMorgan's earnings per share (EPS) in 2022 was $12.09.
=== LLM Response ===
<thinking>
The annual report query provided the missing eps parameter value of $12.09. We
now have both required parameters to calculate the price to earnings ratio:
- price: $134.10
- eps: $12.09
</thinking>
=== Calling Function ===
Calling function: price_earning_ratio with args: {"price": 134.1, "eps": 12.09}
=== Function Output ===
11.09181141439206
=== LLM Response ===
Based on JPMorgan's stock price of $134.10 and earnings per share of $12.09 in
2022, the price to earnings (P/E) ratio was 11.09.
assistant: Based on JPMorgan's stock price of $134.10 and earnings per share of
$12.09 in 2022, the price to earnings (P/E) ratio was 11.09.
```