AI Conclave

Online

AIOT105

Build application using advanced RAG methods and validate using different evaluation mechanism

Shailesh Shivakumar

Senior Solutions Architect
AWS India

Agenda

- RAG introduction
- Challenges with Normal RAG
- Advanced RAG – Parsing
- Advanced RAG - Chunking
- Advanced RAG – Query reformulation
- RAG evaluation
- Demo

RAG introduction

What is Retrieval Augmented Generation?

Retrieval

Fetches the relevant content from the external knowledge base or data sources based on a user query

Augmentation

Adding the retrieved relevant context to the user prompt, which goes as an input to the foundation model

Generation

Response from the foundation model based on the augmented prompt

RAG use cases

Improved content quality

E.g., helps in reducing hallucinations and connecting with recent knowledge including enterprise data

Contextual chatbots and question answering

E.g., enhance chatbot capabilities by integrating with real-time data

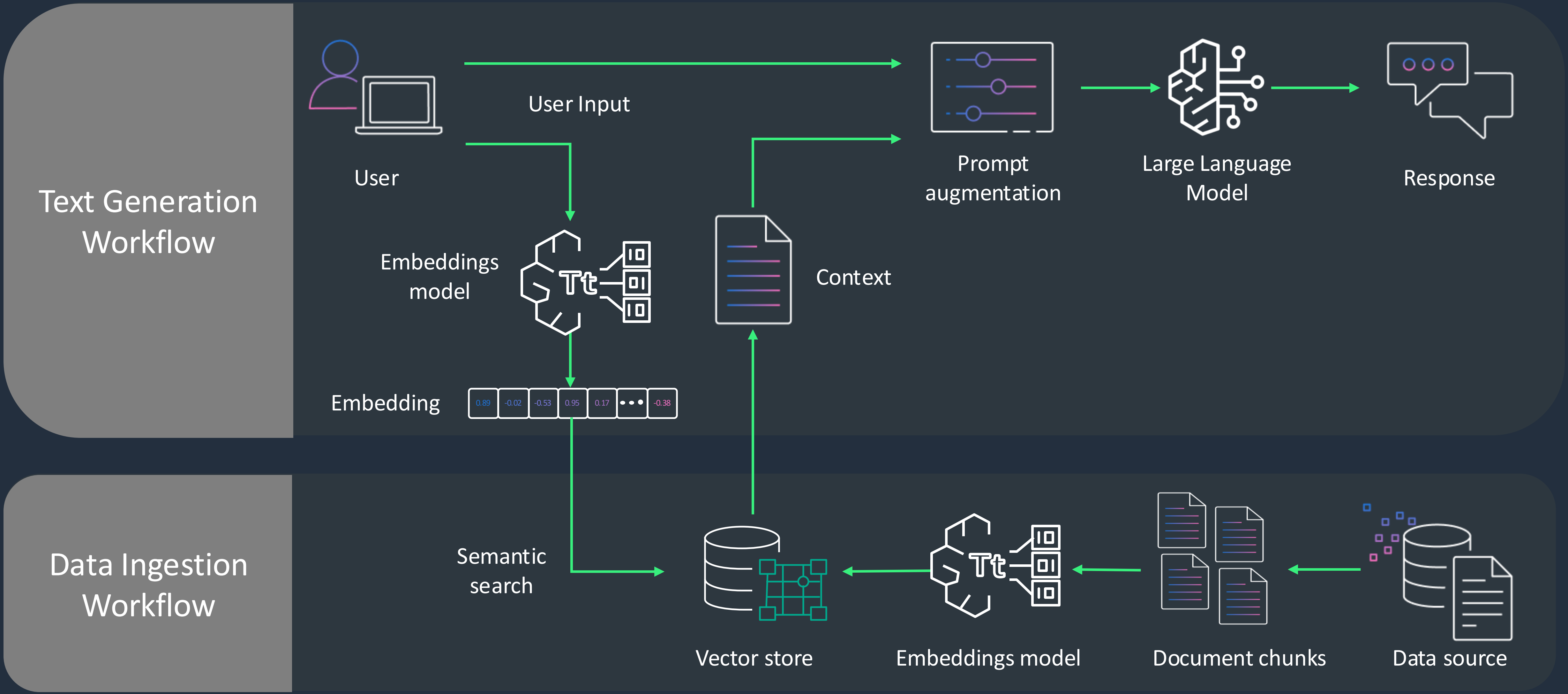
Personalized search

E.g., searching based on user previous search history and persona

Real-time data summarization

E.g., retrieving and summarizing transactional data from databases,
or API calls

RAG in Action



Text Generation Workflow

User

User Input

Embeddings model

Embedding

Context

Prompt augmentation

Large Language Model

Response

Data Ingestion Workflow

Semantic search

Vector store

Embeddings model

Document chunks

Data source

Knowledge Bases for Amazon Bedrock

Gives FMs and agents contextual information from your private data sources for Retrieval Augmented Generation (RAG) to deliver more relevant, accurate, and customized responses.

Fully managed support for end-to-end RAG workflow

Securely connect FMs and agents to data sources

Easily retrieve relevant data and augment prompts

Provide source attribution

RAG challenges

Challenges with Regular RAG Approach

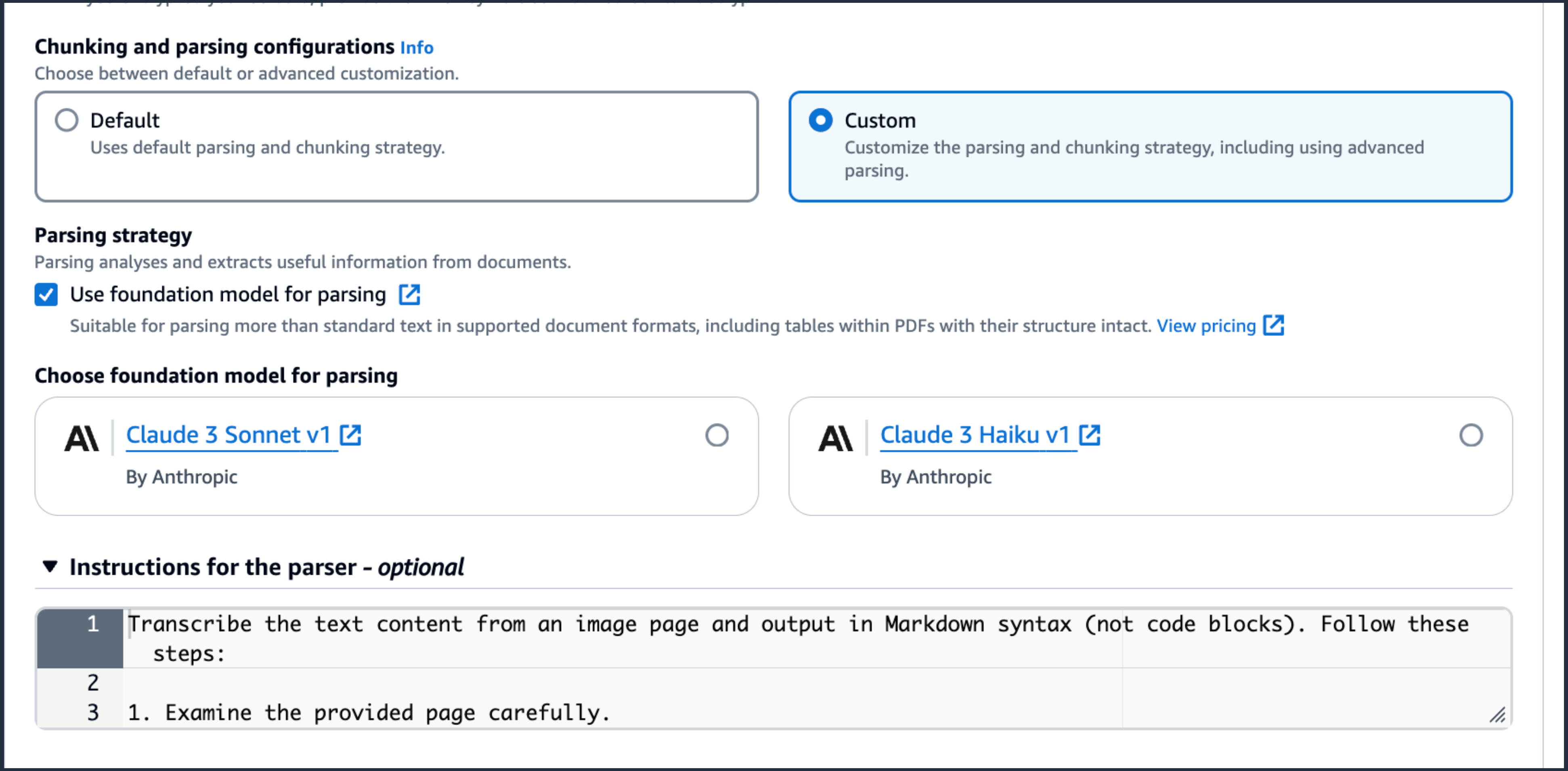
- **Parsing complex structures** – Entities nested tables, images, graphs are not parsed accurately.
- **Use relevant data from large dataset** – As the dataset grows, only the configured set of top results are used for context impacting accuracy.
- **Responding to complex questions** – A compound question (such as ones with different sub-questions) pose challenges to the RAG solution.
- **Retrieve right context with complex document structures** – Few documents such as legal documents or technical manuals have semantically related sections in different pages impacting the overall accuracy.
- **Generate complete, correct and grounded response** without any hallucination.

Workarounds

- **Programatic parsing** – Use different service (such as Amazon Textract) to extract the table content into .md file
- **Programmatic split of compound questions** – Programmatically split the question or reformulate the question and get the responses and programmatically aggregate the response.
- **Programatic chunking** – Programmatically create custom chunks of the documents and mange them in vector store.

Advanced RAG - Parsing

Advanced RAG with Amazon Bedrock Knowledge Bases



Chunking and parsing configurations Info

Choose between default or advanced customization.

○ Default
Uses default parsing and chunking strategy.

● Custom
Customize the parsing and chunking strategy, including using advanced parsing.

Parsing strategy

Parsing analyses and extracts useful information from documents.

☑ Use foundation model for parsing

Suitable for parsing more than standard text in supported document formats, including tables within PDFs with their structure intact. View pricing

Choose foundation model for parsing

○ Claude 3 Sonnet v1
By Anthropic

○ Claude 3 Haiku v1
By Anthropic

▼ **Instructions for the parser - *optional***

```
Transcribe the text content from an image page and output in Markdown syntax (not code blocks). Follow these
  steps:

1. Examine the provided page carefully.
```

Advanced RAG - Chunking

Chunking strategy

What

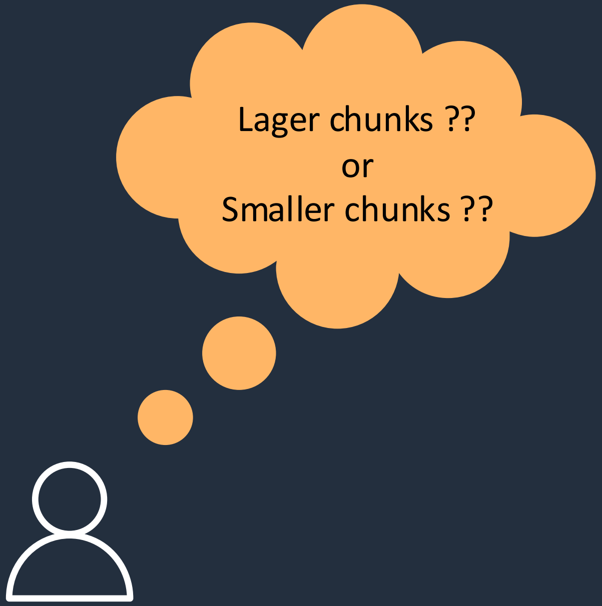
Chunking: Segmenting text into meaningful, concise chunks

Why

Enables semantic text embedding

How

Effectiveness of chunking strategy largely depends on **quality and structure of the chunks**.



Optimal chunk size balances
relevant context and speed

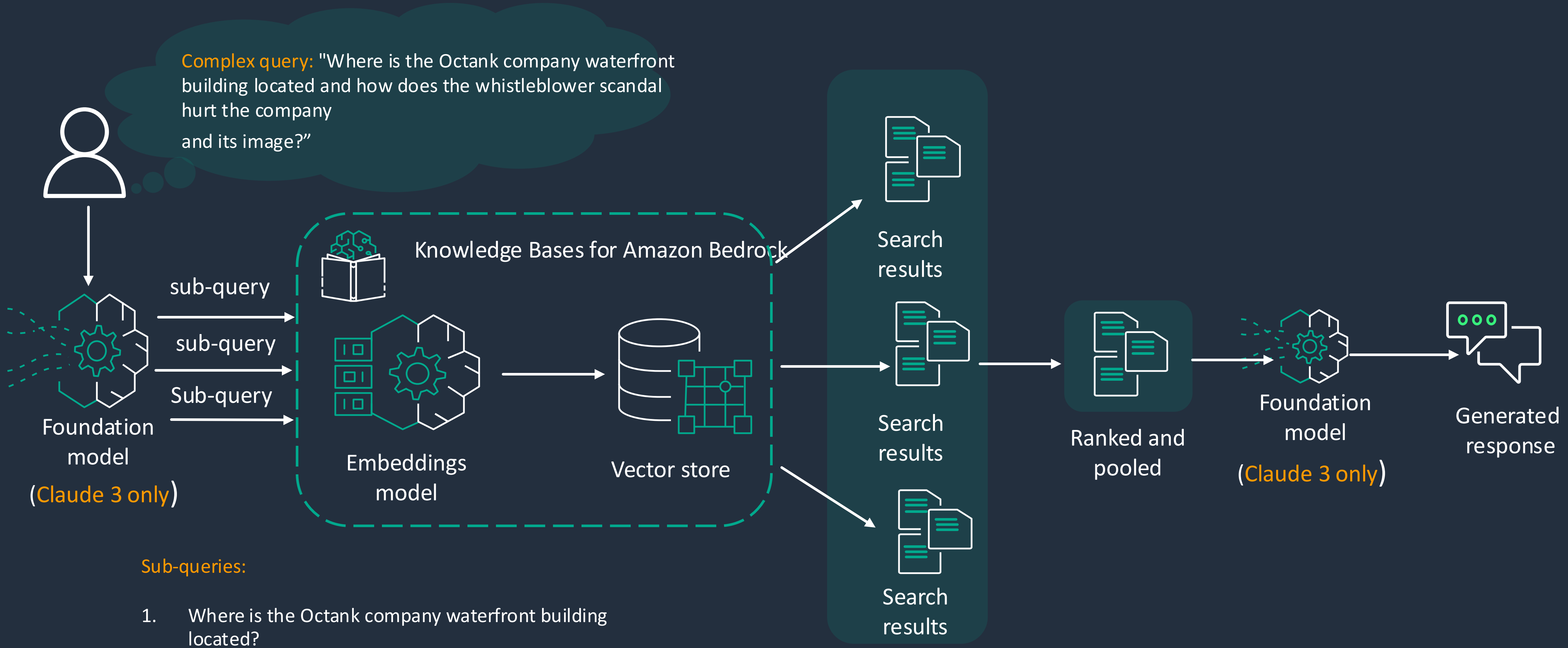
Overview of different chunking approaches

Type	Description	Pros	Cons	Remarks
Fixed chunking	- Fixed character count division - Recursive character text splitting	Quick and efficient	Lacks context awareness	Data is relatively uniform in length and structure.
Semantic chunking	- Meaningful and complete chunks based on semantic similarity	Better retrieval quality		Suitable for lengthy documents that have related sections.
Hierarchical chunking	- Organize chunks in a structured manner - Nodes with summaries, parent-child relationships	Improve retrieval efficiency and reliability	Require domain-specific or problem-specific expertise	When required context is split across multiple different documents

Advanced RAG – Query reformulation

RAG API - Query reformulation

New



Complex query: "Where is the Octank company waterfront building located and how does the whistleblower scandal hurt the company

and its image?”

Sub-queries:

1. Where is the Octank company waterfront building located?
2. What is the whistleblower scandal involving Octank company?
3. How did the whistleblower scandal affect Octank company's reputation and public image?

RAG evaluation

Special challenges with RAG evaluation

- Use relevant data from your knowledge base
- Retrieve the right context from documents
- Generate a correct, complete, and grounded answer minimizing hallucinations
- Iteratively improve your RAG system and compare across changes
- Evaluate biases, safety, and trust

Public Preview

RAG evaluation on Amazon Bedrock Knowledge Bases

Evaluate your full Amazon Bedrock Knowledge Base stack to optimize your RAG application

1. Bring your own datasets for tailored results
2. Evaluate retrieval alone or retrieval + generation with a choice of LLM-as-a-judge
3. Built-in metrics for quality and responsible AI, compatible with Amazon Bedrock Guardrails
4. Compare across multiple evaluation jobs
5. Get results in just a few clicks

RAG evaluation input data format

Input dataset contains 2 things in JSONL format

1. Prompt
2. Optional golden ground truth

```
{
        "conversationTurns": [{
                "referenceResponses": [{
                        "content": [{
                                "text": "This is a reference response"
                        }]
                }],
                "prompt": {
                        "content": [{
                                "text": "This is a prompt"
                        }]
                }
        }]
}
```

Retrieve jobs have `referenceContexts`

Choice of evaluation metrics

Retrieval

Context Coverage

Context Relevance

Retrieval + Generation

Correctness

Completeness

Helpfulness

Logical coherence

Faithfulness

Responsible AI

Harmfulness

Stereotyping

Refusal

How correctness works

Example input

prompt: What is the capital of Spain?

referenceResponse: Madrid

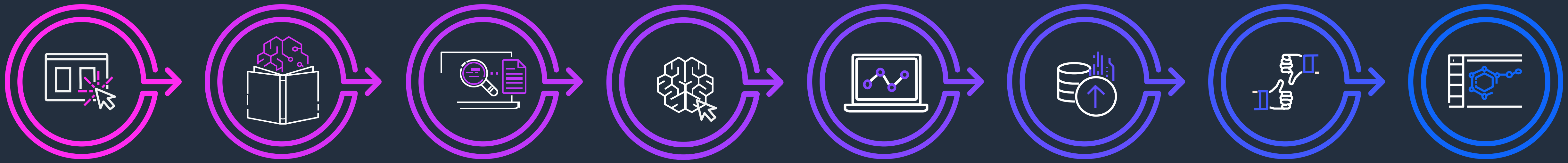
Model response: Barcelona

Judge prompt (simplified)

```
You are a helpful assistant…

You are given a question, a candidate
response from an LLM, and reference response.

Your task is to check if the candidate
response is correct compared to the reference
response…

Here is the actual task:
Question:  {prompt}
Reference Response: {referenceResponse}
Candidate Response: {Model response}

Explain your response, followed by your
evaluation:
2) Correct
1) Partially correct
0) Incorrect
```

Note: Correctness can be with or without ground truth

How RAG evaluation works with Knowledge Bases



Choose evaluator model

Choose your Knowledge Base

Choose to evaluate retrieval only or retrieve + generate

Choose your generator model

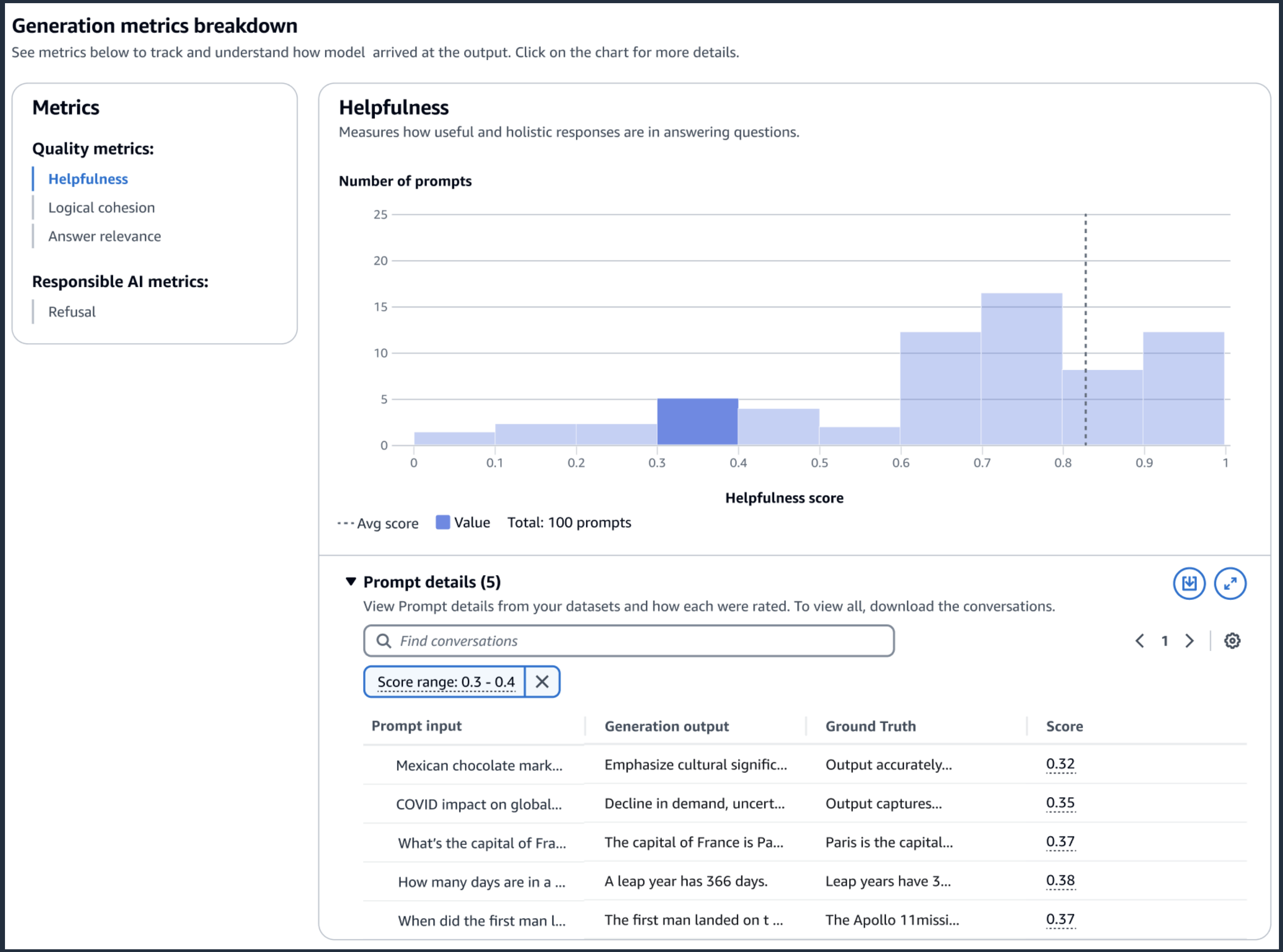
Choose your metrics

Upload your prompt dataset

Inference and evaluation

View results

Get results in a few clicks



Generation metrics breakdown

See metrics below to track and understand how model arrived at the output. Click on the chart for more details.

Metrics

Quality metrics:

- Helpfulness
- Logical cohesion
- Answer relevance

Responsible AI metrics:

- Refusal

Helpfulness

Measures how useful and holistic responses are in answering questions.

▼ Prompt details (5)

View Prompt details from your datasets and how each were rated. To view all, download the conversations.

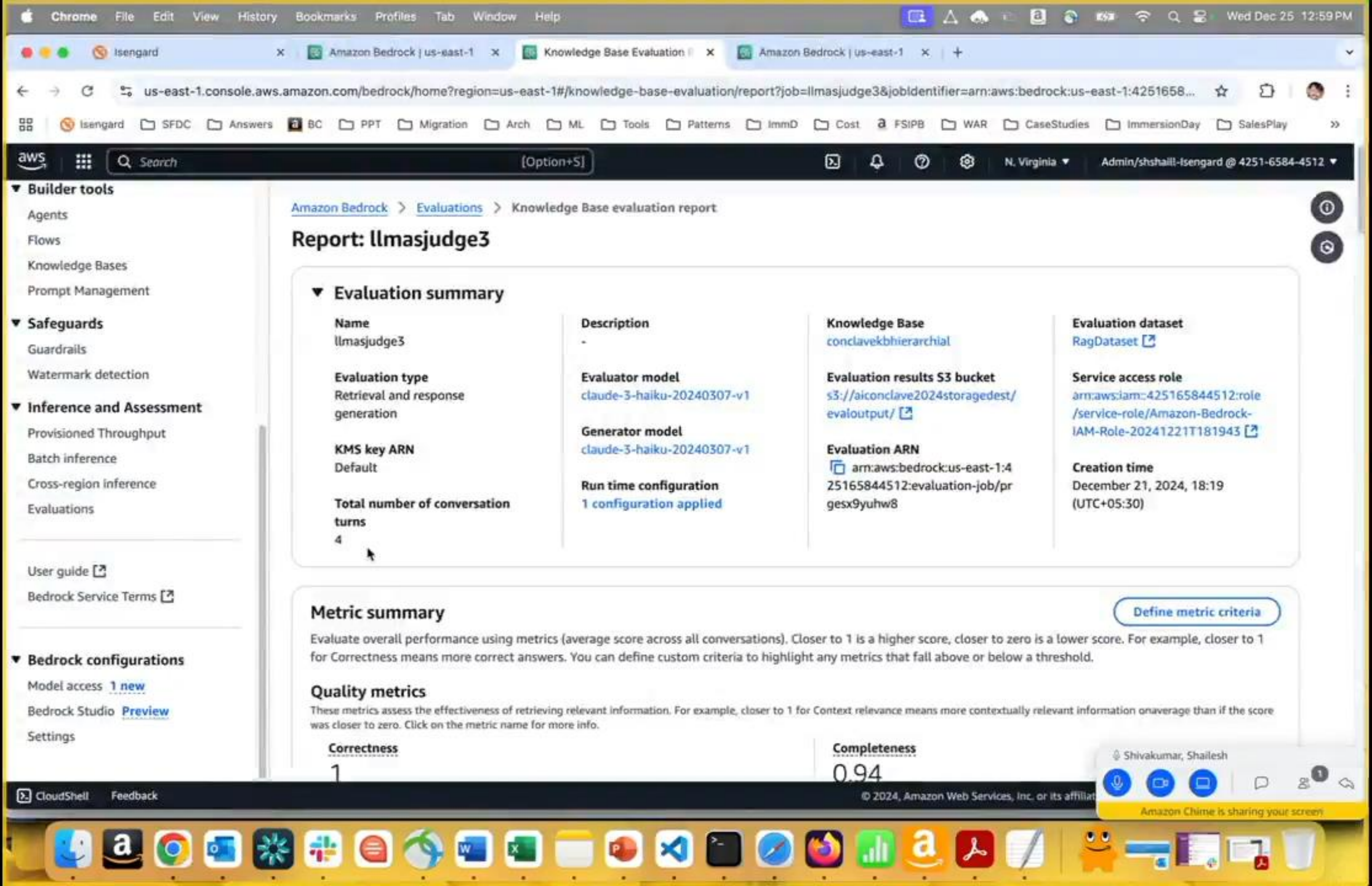
Find conversations

Score range: 0.3 - 0.4 ✕

Prompt input	Generation output	Ground Truth	Score
Mexican chocolate mark...	Emphasize cultural signific...	Output accurately...	0.32
COVID impact on global...	Decline in demand, uncert...	Output captures...	0.35
What's the capital of Fra...	The capital of France is Pa...	Paris is the capital...	0.37
How many days are in a ...	A leap year has 366 days.	Leap years have 3...	0.38
When did the first man l...	The first man landed on t ...	The Apollo 11missi...	0.37

- Simple to read scores
- See distributions visually
- See ratings explanations

Demo


Chrome File Edit View History Bookmarks Profiles Tab Window Help
Wed Dec 25 12:59 PM
Isengard
Amazon Bedrock | us-east-1
Knowledge Base Evaluation
Amazon Bedrock | us-east-1
us-east-1.console.aws.amazon.com/bedrock/home?region=us-east-1#/knowledge-base-evaluation/report?job=llmasjudge3&jobIdentifier=arn:aws:bedrock:us-east-1:4251658...
Isengard SFDC Answers BC PPT Migration Arch ML Tools Patterns ImmD Cost FSIPB WAR CaseStudies ImmersionDay SalesPlay
aws
Search [Option+S]
N. Virginia
Admin/shshaill-Isengard @ 4251-6584-4512
Builder tools
Agents
Flows
Knowledge Bases
Prompt Management
Safeguards
Guardrails
Watermark detection
Inference and Assessment
Provisioned Throughput
Batch inference
Cross-region inference
Evaluations
User guide
Bedrock Service Terms
Bedrock configurations
Model access 1 new
Bedrock Studio Preview
Settings
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliat
Shivakumar, Shailesh
Amazon Chime is sharing your screen

Amazon Bedrock > Evaluations > Knowledge Base evaluation report

Report: llmasjudge3

Evaluation summary

Name llmasjudge3	**Description** -	**Knowledge Base** conclavekbhierarchial	**Evaluation dataset** RagDataset
Evaluation type Retrieval and response generation	**Evaluator model** claude-3-haiku-20240307-v1	**Evaluation results S3 bucket** s3://aiconclave2024storagedest/evaloutput/	**Service access role** arn:aws:iam::425165844512:role/service-role/Amazon-Bedrock-IAM-Role-20241221T181943
KMS key ARN Default	**Generator model** claude-3-haiku-20240307-v1	**Evaluation ARN** arn:aws:bedrock:us-east-1:425165844512:evaluation-job/prgesx9yuhw8	**Creation time** December 21, 2024, 18:19 (UTC+05:30)
Total number of conversation turns 4	**Run time configuration** 1 configuration applied		

Metric summary

Define metric criteria

Evaluate overall performance using metrics (average score across all conversations). Closer to 1 is a higher score, closer to zero is a lower score. For example, closer to 1 for Correctness means more correct answers. You can define custom criteria to highlight any metrics that fall above or below a threshold.

Quality metrics

These metrics assess the effectiveness of retrieving relevant information. For example, closer to 1 for Context relevance means more contextually relevant information onaverage than if the score was closer to zero. Click on the metric name for more info.

Correctness	Completeness
1	0.94

Thank you!

Shailesh Shivakumar

Senior Solutions Architect
AWS India