Part 3: Talk to your slide deck (Multimodal RAG) using foundation models (FMs) hosted on Amazon Bedrock

*Archana Inapudi*, *Manju Prasad*, *Amit Arora*, *Antara Raisa*

In the “Talk to your slide deck” blog series, we share two approaches to gain insights on multimodal data like text, images, and charts. In [part 1](https://aws.amazon.com/blogs/machine-learning/talk-to-your-slide-deck-using-multimodal-foundation-models-hosted-on-amazon-bedrock-and-amazon-sagemaker-part-1/), we presented an “embed first, infer later” solution that uses [Amazon Titan Multimodal Embeddings](https://docs.aws.amazon.com/bedrock/latest/userguide/titan-multiemb-models.html) model to convert individual slides from a slide deck into embeddings. We stored the embeddings in a vector database and then used the [Large Language-and-Vision Assistant (LLaVA 1.5-7b)](https://llava-vl.github.io/) model to generate text responses to user questions based on the most similar slide retrieved from the vector database. Part 1 uses AWS services including [Amazon Bedrock](https://aws.amazon.com/bedrock/), [Amazon SageMaker](https://aws.amazon.com/sagemaker/), and [Amazon OpenSearch Serverless](https://aws.amazon.com/opensearch-service/features/serverless/).

In [part 2](https://aws.amazon.com/blogs/machine-learning/talk-to-your-slide-deck-using-multimodal-foundation-models-hosted-on-amazon-bedrock-and-amazon-sagemaker-part-2/), we demonstrated a different approach “infer first, embed later”. We used [Anthropic Claude 3 Sonnet](https://aws.amazon.com/bedrock/claude/) model to generate text descriptions for each slide in the slide deck. These descriptions are then converted into text embeddings using [Amazon Titan Text Embeddings](https://docs.aws.amazon.com/bedrock/latest/userguide/titan-embedding-models.html) model and stored in a vector database. Then we used the Claude 3 Sonnet model to generate answers to user questions based on the most relevant text description retrieved from the vector database.

In this post, we evaluate the results from both approaches using ground truth provided by [SlideVQA](https://github.com/nttmdlab-nlp/SlideVQA), an open-source visual question answering dataset. You can test both approches and evaluate the results to find the best fit for your datasets. The code for this series is available in the [GitHub](https://github.com/aws-samples/multimodal-rag-on-slide-decks/tree/main) repo.

## Solution overview

Please refer to Solution overview and design in Parts 1 & 2 of the series.

## Comparison

SlideVQA is a collection of publicly available slide decks, each composed of multiple slides (in jpg format) and questions based on the information in the slide decks. It allows a system to select a set of evidence images and answer the question. We use SlideVQA as the single source of truth to compare the results.

### Process

This post will follow the process depicted in the diagram below.

|  |
| --- |
| Figure 1: Process flow |

SlideVQA is a large question answering dataset comprising of thousands of slide decks. We selected 100 random questions to create a [sample dataset](qa.jsonl) to test solutions from Part 1 and Part 2.

The responses to the questions in the sample dataset are concise in as few words as possible, see examples below.

"question": "What is the process by which the breaking of hydrogen bonds allows water to change from the liquid phase into the gaseous phase which has reached equilibrium with the liquid surface said to have done?"  
"answer": "reached saturation"

The responses from large language models (LLMs) are quite verbose.

According to the information provided in the images, the process by which the breaking of hydrogen bonds allows water to change from the liquid phase into the gaseous phase that has reached equilibrium with the liquid surface is said to have reached saturation.  
  
The key points are:  
  
1. Evaporation involves the breaking of hydrogen bonds that hold water molecules together in the liquid phase, allowing them to transition into the gaseous (vapor) phase.  
  
2. Only the fastest moving water molecules with enough energy can overcome the hydrogen bonding and evaporate into the vapor phase.  
  
3. The evaporation process that has reached equilibrium with the liquid surface, where the vapor pressure is balanced with the evaporation rate, is described as having reached saturation.  
  
So in summary, the breaking of hydrogen bonds provides the mechanism for water molecules to gain enough energy to escape the liquid phase as vapor, and when this vapor has reached equilibrium with the liquid surface, it is said to have reached saturation.

We updated the prompts in each approach to provide short instead of verbose responses. This helped to match the output context length to the ground truth responses in the sample dataset.

Below sections will briefly discuss the solutions and dive into the evaluation and pricing for each approach.

#### Approach 1 (embed first, infer later)

Slide decks are converted into pdf images, one per slide, and embedded using the Titan Multimodal Embeddings model, resulting in a vector embedding of 1,024 dimensions. The embeddings are stored in OpenSearch Serverless index which serves as the vector store for our RAG solution. The embeddings are ingested via [Amazon OpenSearch Ingestion Pipeline](https://docs.aws.amazon.com/opensearch-service/latest/developerguide/ingestion.html) (OSI).

Each question is converted into embeddings using the Titan Multimodal Embeddings model and an OpenSearch vector search is performed using these embeddings. We performed a k-nearest neighbor (knn) search to retrieve the most relevant embedding matching the question. The metadata of the response from OpenSearch index contains a path to the image corresponding to the most relevant slide.

A prompt is created by combining the question and the image path and sent to Claude 3 Sonnet to respond to the question with a concise answer.

Note: We used Claude 3 Sonnet instead of LLaVA 1.5-7b as mentioned in the solution for Part 1. The approach remains the same, “embed first, infer later”, just the model that compiles the final response is changed for simplicity and comparability between approaches.

##### Evaluation

A response for each question in the dataset is [recorded](./notebooks/responses-appr1.json) and compared to the ground truth provided by SlideVQA.

Approach 1 retrieved a response for 78% of the questions on a dataset of 100 questions achieving a 50% accuracy on the final responses.

#### Approach 2 (infer first, embed later)

Slide decks are converted into pdf images, one per slide, and passed to the Claude 3 Sonnet model to generate a text description. The description is sent to the Titan Text Embeddings model to generate vector embeddings with 1,536 dimensions. The embeddings are ingested into OpenSearch Serverless index via OSI pipeline.

Each question is converted into embeddings using the Titan Text Embeddings model and an OpenSearch vector search is performed using these embeddings. We performed a k-nearest neighbor (knn) search to retrieve the most relevant embedding matching the question. The metadata of the response from OpenSearch index contains the image description corresponding to the most relevant slide.

We create a prompt with the question and the image description and pass it to Claude 3 Sonnet to receive a precise answer.

##### Evaluation

A response for each question in the dataset is [recorded](./notebooks/responses-appr2.json) and compared to the ground truth provided by SlideVQA.

With approach 2, we received 44% accuracy on final responses with 75% of the questions retrieving a response out of the 100 questions in the sample dataset.

#### Analysis of results

In our testing, both approaches produced <=50% matching results to the questions in the sample dataset. The sample dataset contains a random selection of slide decks covering a wide variety of topics including retail, healthcare, academic, technology, personal, travel etc. So for a generic question like “What are examples of tools that can be used?” that lacks additional context, the nearest match could retrieve responses from a variety of topics leading to inaccurate results.

The embeddings were all ingested into a single index in OpenSearch. Embeddings are vectors which are essentially compressions of the “meaning” behind text into dimensions, 1,024 or 1,536 in this case. There is some information loss when we compress information into a single vector.

One of the solutions is to retrieve more results (increase the k value) and reorder them to keep the most relevant ones, this technique is called reranking. We will share additional ideas on ways to improve the accuracy of the results in the conclusion below.

The [final prompt](./prompts/) to Claude 3 Sonnet in our analysis included instructions to provide a concise answer in as few words as possible to be able to compare with the ground truth. We assume your responses will depend on your prompts to the LLM.

### Pricing

Pricing is dependent on the modality, provider, and model used. Please refer to the detailed public pricing for Amazon Bedrock [here](https://aws.amazon.com/bedrock/pricing/). We use the On-Demand and Batch pricing mode in our analysis that allow you to use FMs on a pay-as-you-go basis without having to make any time-based term commitments. For text-generation models, you are charged for every input token processed and every output token generated. For embeddings models, you are charged for every input token processed.

The below table shows price per question for each approach.

Note: We calculated the average number of input and output tokens based on our sample dataset for us-east-1, pricing may vary for your datasets and Amazon region used.

|  |  | **Approach 1** |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | **Input tokens** |  |  | **Output tokens** |  |  |
| **Model** | **Description** | **Price per 1000 tokens / Price per input image** | **# of tokens** | **Price** | **Price per 1000 tokens** | **# of tokens** | **Price** |
| Titan Multimodal Embeddings | Slide/image embedding | $0.000060 | 1 | $0.0000000600 | 0.000000 | 0 | $0.0000000000 |
| Titan Multimodal Embeddings | Question embedding | $0.000800 | 20 | $0.0000160000 | 0.000000 | 0 | $0.0000000000 |
| Claude 3 Sonnet | Final response | $0.003000 | 700 | $0.0021000000 | 0.015000 | 8 | $0.0001200000 |
| Cost per input/output |  |  |  | $0.0021160600 |  |  | $0.0001200000 |
| **Total cost per question** |  |  |  |  |  |  | **$0.0022360600** |
|  |  |  |  |  |  |  |  |
|  |  | **Approach 2** |  |  |  |  |  |
|  |  | **Input tokens** |  |  | **Output tokens** |  |  |
| Model | **Description** | **Price per 1000 tokens / Price per input image** | **# of tokens** | **Price** | **Price per 1000 tokens** | **# of tokens** | **Price** |
| Claude 3 Sonnet | Slide/image description | $0.003000 | 4523 | 0.0135690000 | 0.015000 | 350 | $0.0052500000 |
| Titan Text Embeddings | Slide/image description embedding | $0.000100 | 350 | 0.0000350000 | 0.000000 | 0 | $0.0000000000 |
| Titan Text Embeddings | Question embedding | $0.000100 | 20 | 0.0000020000 | 0.000000 | 0 | $0.0000000000 |
| Claude 3 Sonnet | Final response | $0.003000 | 700 | 0.0021000000 | 0.015000 | 8 | $0.0001200000 |
| Cost per input/output |  |  |  | 0.0157060000 |  |  | $0.0053700000 |
| **Total cost per question** |  |  |  |  |  |  | **$0.0210760000** |

## Conclusion

In this series, we explored ways to use the power of multimodal FMs such as Titan Multimodal Embeddings, Titan Text Embeddings, and Claude 3 Sonnet models to discover new information and uncover new perspectives on content in slide decks. We encourage you to explore different Claude models available on Bedrock.

With Generative AI being a fast moving space, there are several ways to improve the results and/or approach the problem. We are exploring performing a hybrid search and adding search filters by extracting entities from the question to improve the results. Lookout for a blog on “Talk to your PDF files (Enhanced Multimodal RAG) using foundation models (FMs) hosted on Amazon Bedrock” that will explore these concepts in detail.

Portions of this code are released under the Apache 2.0 License as referenced here: https://aws.amazon.com/apache-2-0/

## Author bio

Archana Inapudi is a Senior Solutions Architect at AWS, supporting a strategic customer. She has over a decade of cross-industry expertise leading strategic technical initiatives. Archana is an aspiring member of the AIML technical field community at AWS. Prior to joining AWS, Archana led a migration from traditional siloed data sources to Hadoop at a health care company. She is passionate about using technology to accelerate growth, provide value to customers, and achieve business outcomes.

Manju Prasad is a Senior Solutions Architect at Amazon Web Services. She focuses on providing technical guidance in a variety of technical domains, including AI/ML. Prior to joining AWS, she designed and built solutions for companies in the financial services sector and also for a startup. She has worked in all layers of the software stack, ranging from webdev to databases and has experience in all levels of the software development lifecycle. She is passionate about sharing knowledge and fostering interest in emerging talent.

Amit Arora is an AI and ML Specialist Architect at Amazon Web Services, helping enterprise customers use cloud-based machine learning services to rapidly scale their innovations. He is also an adjunct lecturer in the MS data science and analytics program at Georgetown University in Washington D.C.

Antara Raisa is an AI and ML Solutions Architect at Amazon Web Services supporting Strategic Customers based out of Dallas, Texas. She also has previous experience working with large enterprise partners at AWS, where she worked as a Partner Success Solutions Architect for digital native customers.