Learn and Launch
- Generative AI Application -

AGENDA

1. Understand Generative AI
2. Word Embedding
3. Prompt Engineering for LLM and Image Model
4. Build a Generative AI Application

Lecture

Lab

Ayyanar Jeyakrishnan

bit.ly/awsaj

AWS community builder
Machine Learning

ABOUT ME

I started my career as humble Hardware and Networking engineer in 2005 in HCL Infosystem.

As a Multi Cloud Architect, I have around 18+ years of IT Experience, 50+ Industry Certification, I am deeply passionate about designing and implementing scalable and efficient cloud solutions also Machine Learning and Data Platform for enterprises.

Currently, I am working as VP - Principal Engineer in Wells Fargo.

I always love to learn - Unlearn - Relearn, Motivated to Share knowledge with peers, community and learn from them.

Path to Gen AI

Descriptive Analytics

Charts and Graphs - Using BI

Predictive Analytics

Linear Regression
Logistic Regression
Clustering

Predictive AI & Descriptive AI

Word Embeddings

Word2Vec

RNN, LSTM, GRU
CNN, YOLO

Path to Gen AI

Predictive Analytics

Descriptive Analytics

Linear Regression
Logistic Regression
Clustering

Charts and Graphs - Using BI

Path to Gen AI

Descriptive Analytics

Charts and Graphs - Using BI

Predictive Analytics

Linear Regression
Logistic Regression
Clustering

Predictive AI & Descriptive AI

Word Embeddings

Word2Vec

RNN, LSTM, GRU
CNN, YOLO

Path to Gen AI

Descriptive Analytics

Charts and Graphs - Using BI

Predictive Analytics

Linear Regression
Logistic Regression
Clustering

Predictive AI & Descriptive AI

Word Embeddings

Word2Vec

RNN, LSTM, GRU
CNN, YOLO

Generative AI

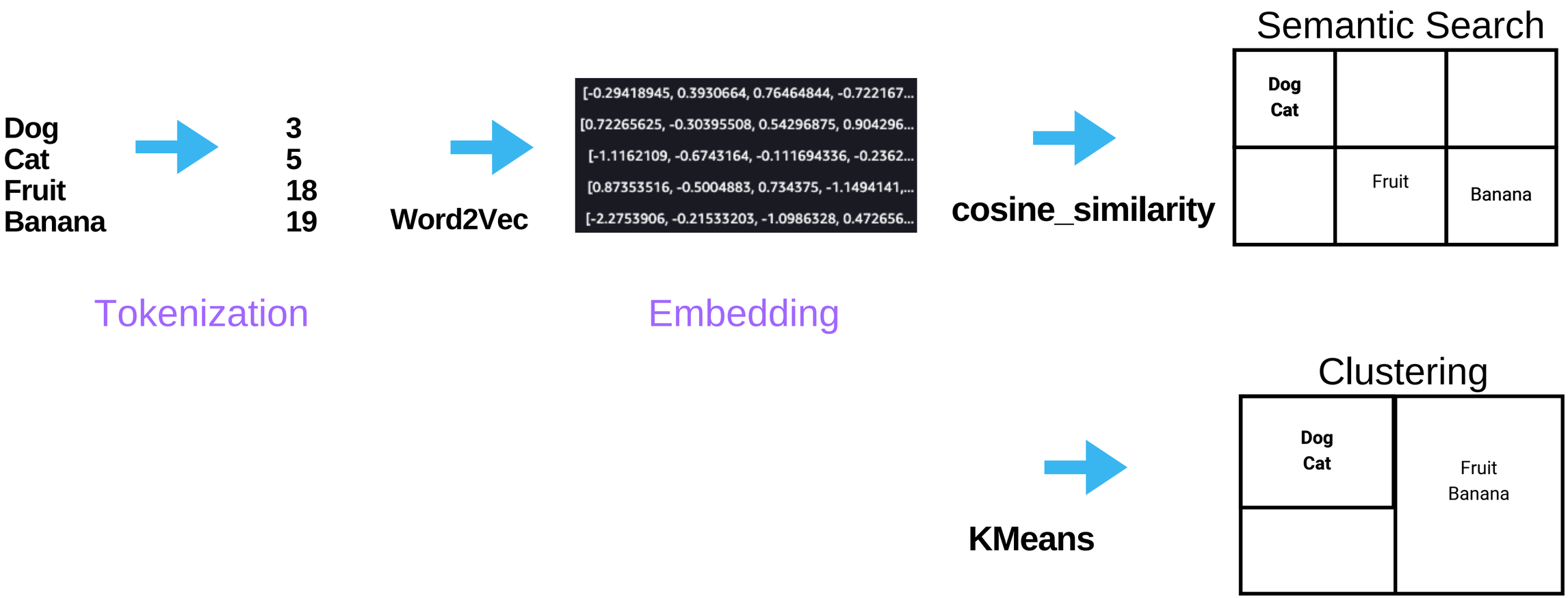
GAN (Gen and Des)

<u>Transformer Models</u>

Falcon, MPT, Dolly, GPT-3.5, Bloom, FLAN-T5, BART and **BERT, LLaMA-2**

<u>Stable Diffusion</u>

Let us Understand Basics on How Computer Learn

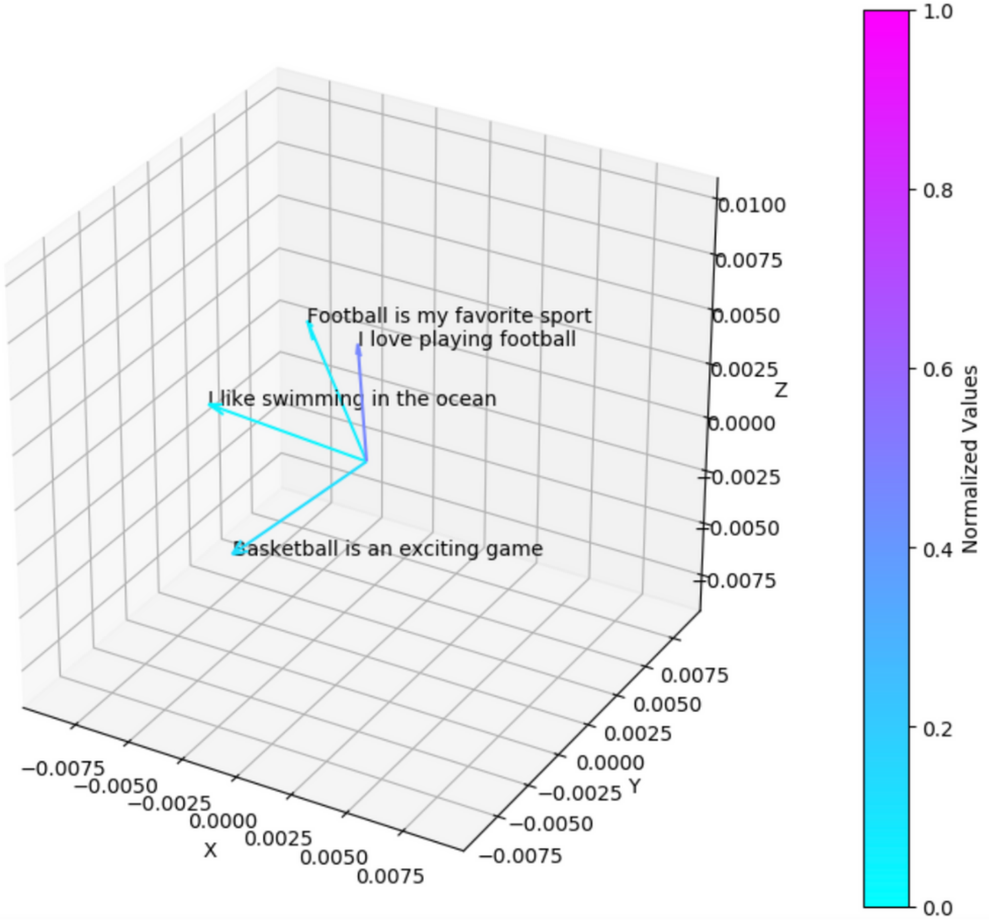


Lab 1 - Word Embedding

```
corpus = [
    "I love playing football",
    "Football is my favorite sport",
    "I enjoy watching football matches",
    "Soccer is popular worldwide"
]
```

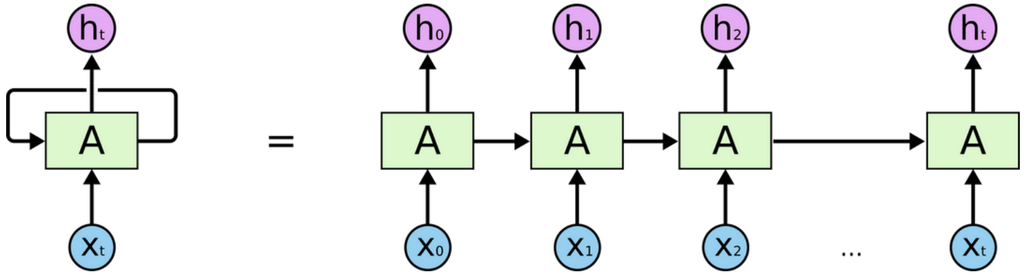
```
Vocabulary size: 15

Word Vocabulary:
is
football
I
worldwide
popular
Soccer
matches
watching
enjoy
sport
favorite
my
Football
playing
love
```

```
Similar words to 'football':
love: 0.16072481870651245
playing: 0.13725270330905914
worldwide: 0.06797593832015991
enjoy: 0.03363519534468651
Soccer: 0.009391184896230698
my: 0.008315940387547016
popular: 0.004503006115555763
Football: -0.003644464071840048
is: -0.010839183814823627
I: -0.023671666160225868
```

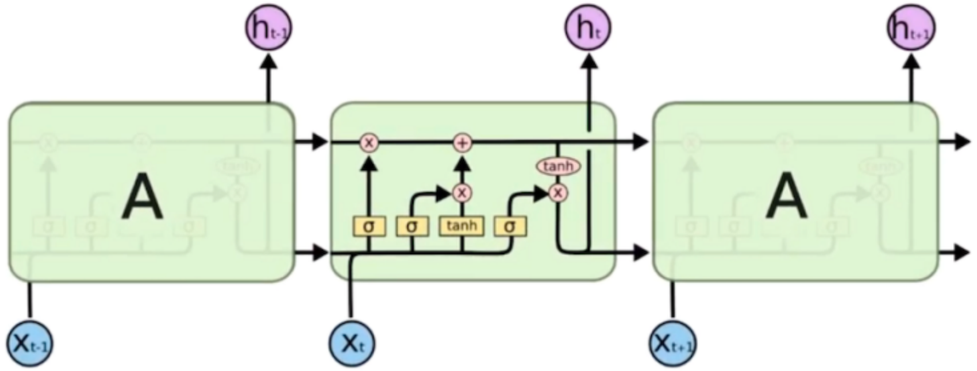


Feed Forward Network - Sequential Models

RNN



LSTM Networks



The repeating module in an LSTM contains four interacting layers.

the challenge

- It’s slow to train.
- Long sequences lead to vanishing gradient or the problem of long-term dependencies. In simple terms, its memory is not that strong when it comes to remembering old connections.

- Vanishing gradient
- Slow training

Transformers - Attention Is All You Need - 2017

The Transformer model architecture Advantage.

- “**Multi Headed Attention**” enabled models to scale the understanding of relationships between words and help to give **context.**
- Efficiently use **Parallel computing**.
- Transfer Learning and Adaptability

  Scalability

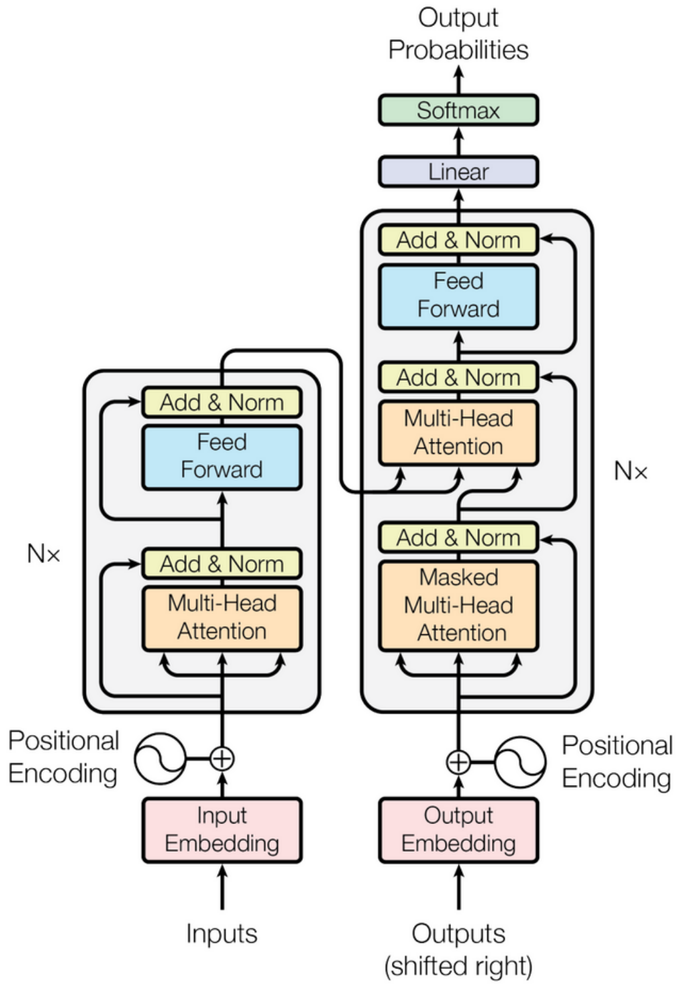
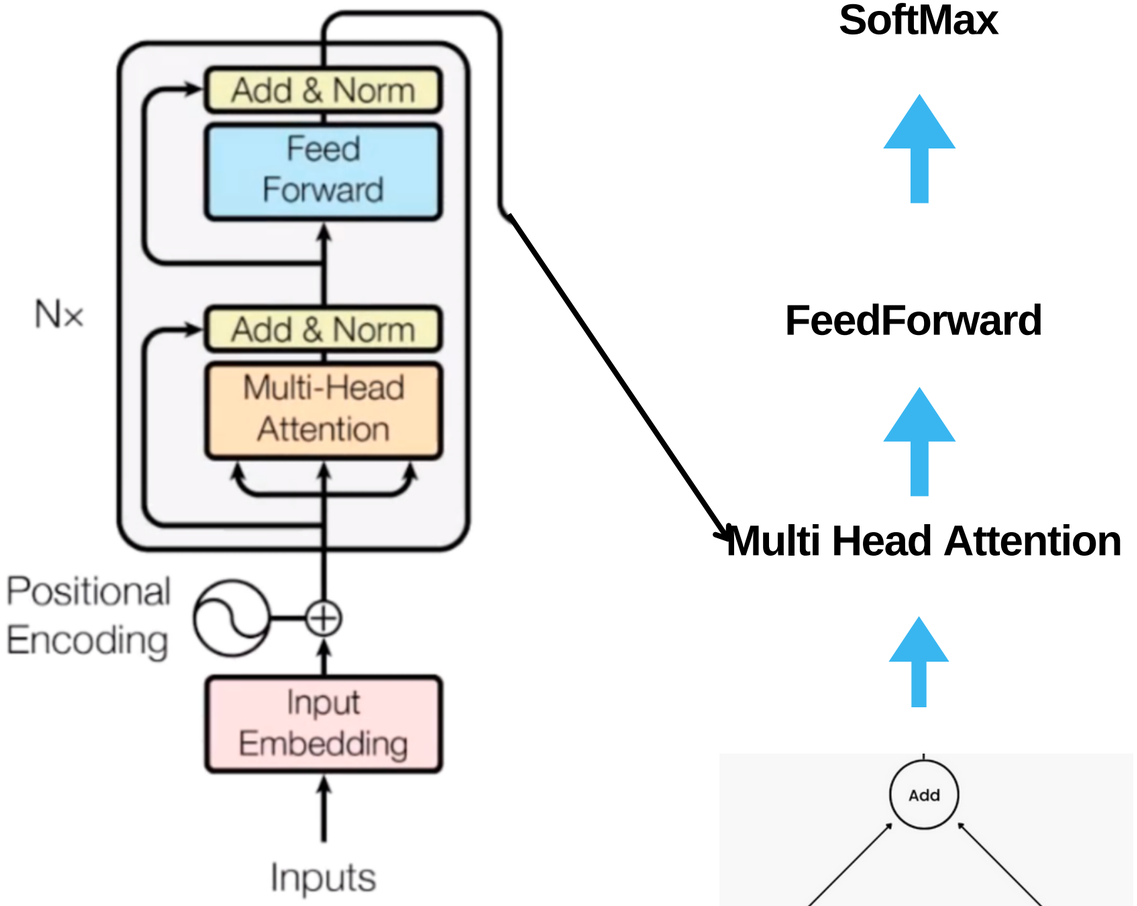


Figure 1: The Transformer - model architecture.

Transformer - Encoder - Intuitive Understanding



SoftMax

FeedForward

Multi Head Attention

Decoder Network

A feed-forward NN is the next step. We can apply a simple feed-forward neural network to each attention vector to transform attention vectors into a form that the next encoder or decoder layer will accept it easily.

Imagine multi-head attention as a multi-tasker. You have a word with you, from which the Transformer deep learning model needs to learn the next word. Multi-head attention performs different parallel computations for the same word to achieve different results. These results are then connected to SoftMax to output the best suitable word.

E.g - 1) I liked the cricket bat purchased from Amazon.in

E.g - 2) I appreciate the prompt delivery by amazon and product description was good, But the product not lived upto my expectation

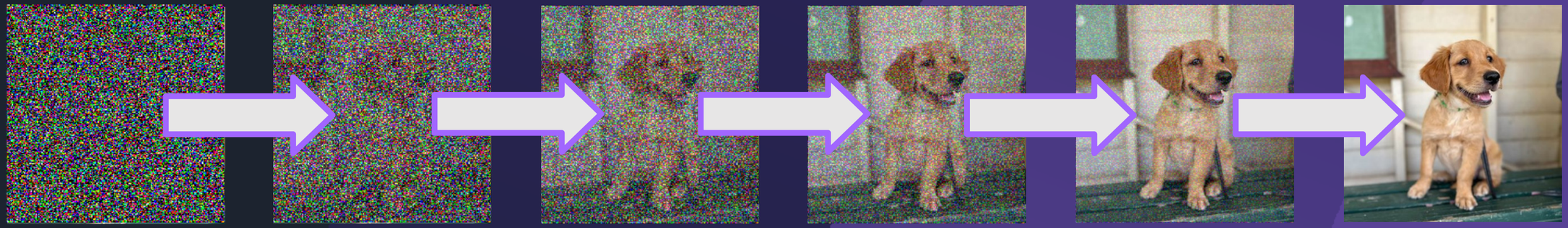
E.g 3) - Magnificent six strike into the crowd

Provide Position + Word2Vec Embedding

Stable Diffusion

Diffusion models.

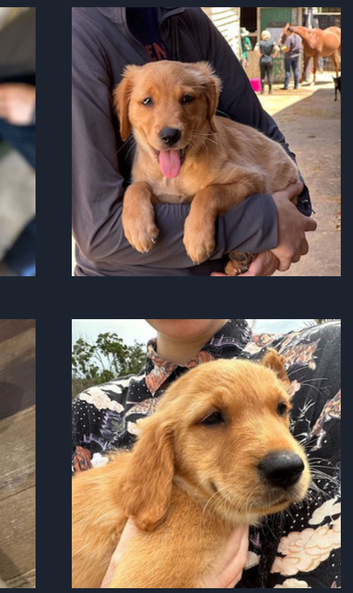
Adds noise and learns how to work backwards to the original image.



Trained model works from random noise to generated an image.

Stable Diffusion Examples

Diffusion models.



the puppy”

Stable Diffusion Training

“Marigold the puppy on a space station”

Foundation models available on SageMaker JumpStart for self-managed access

Publicly available

stability.ai

Models

Text2Image

Upscaling

Tasks

Generate photo-realistic images from text input

Improve quality of generated images

Features

Fine-tuning on SD 2.1 model

alexa

Models

AlexaTM 20B

Tasks

Machine translation

Question answering

Summarization

Annotation

Data generation

Hugging Face

Models

Flan T-5 models (8 variants)

DistilGPT2, GPT2

Bloom models (3 variants)

Tasks

Machine translation

Question answering

Summarization

Annotation

Data generation

Proprietary models ?

co:here

Models

Cohere generate-med

Tasks

Text generation

Information extraction

Question answering

Summarization

LightOn

Models

Lyra-Fr 10B

Tasks

Text generation

Keyword extraction

Information extraction

Question answering

Summarization

Sentiment analysis

Classification

AI21labs

Models

Jurassic-1 Grande 17B

Tasks

Text generation

Long-form generation

Summarization

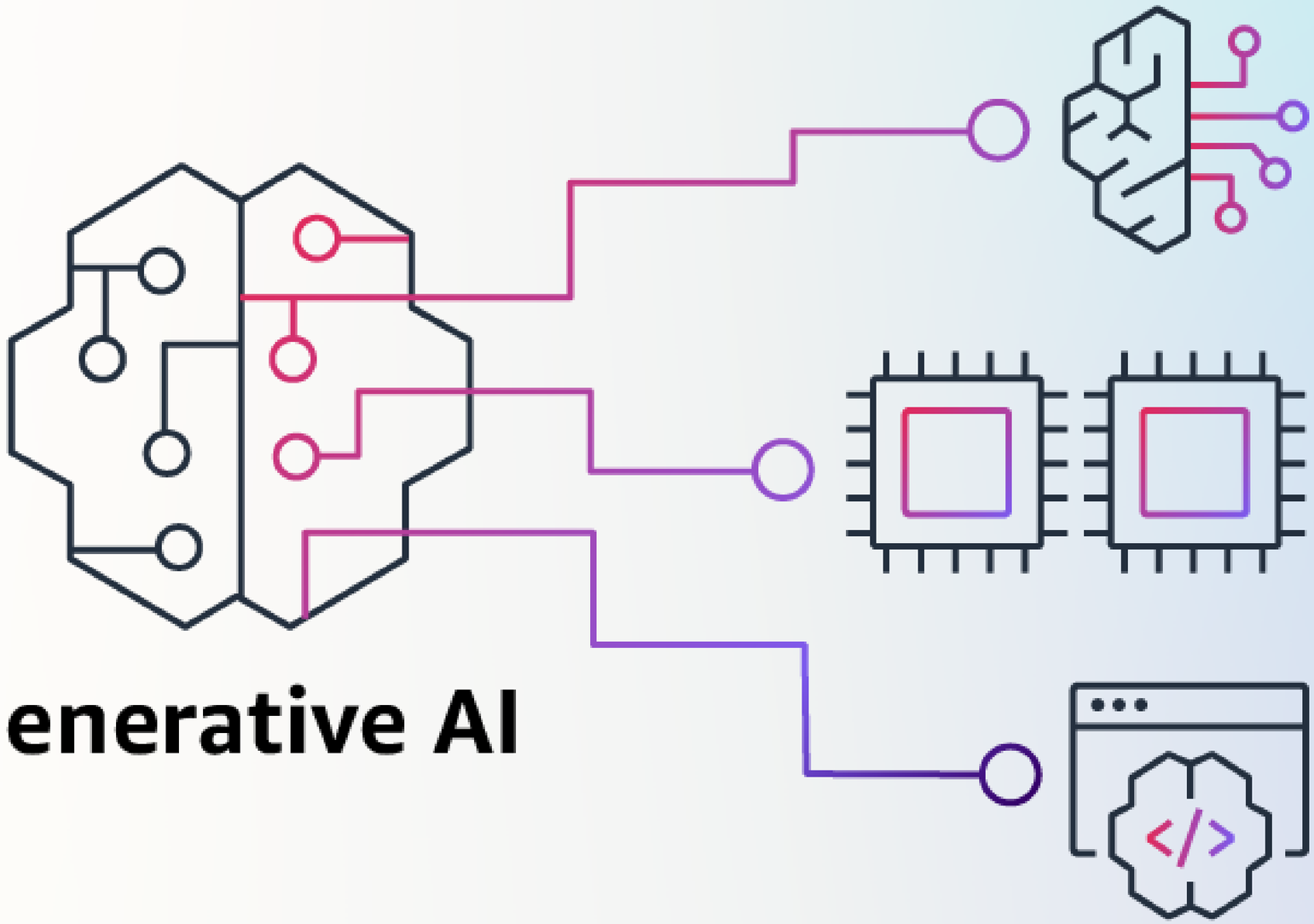
Paraphrasing

Chat

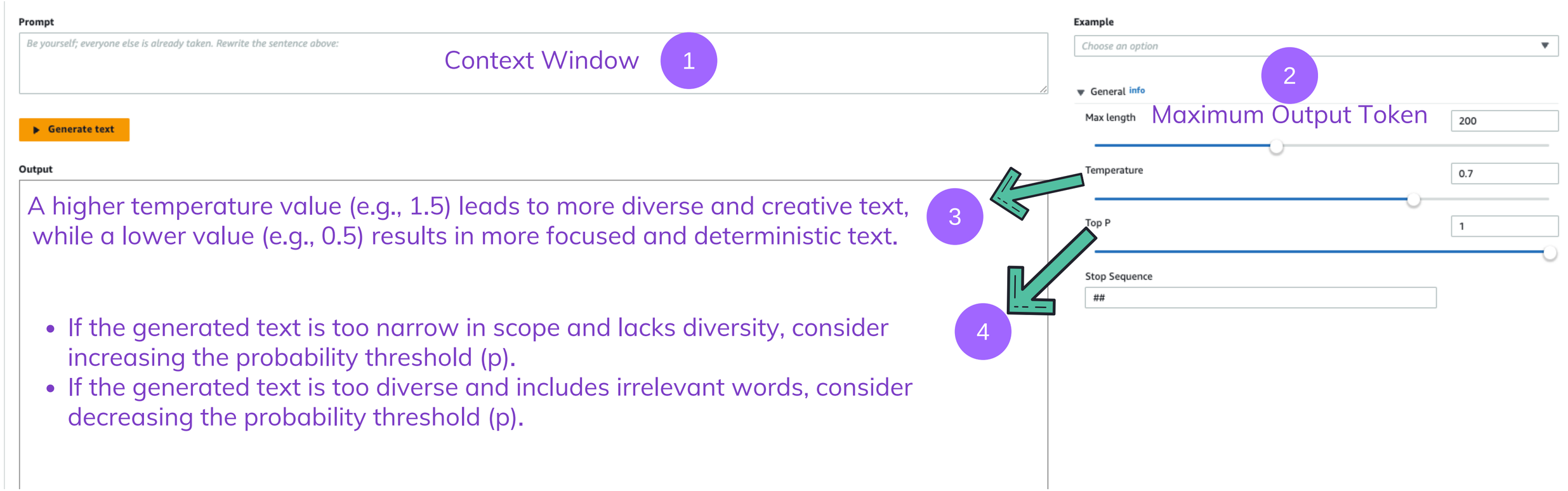
Information extraction

Question answering

Classification


Generative AI
Amazon Bedrock
Amazon EC2 Trn1n
and Amazon EC2 Inf2
Amazon CodeWhisperer

Connect with Large Language Model Easily
- Prompting -

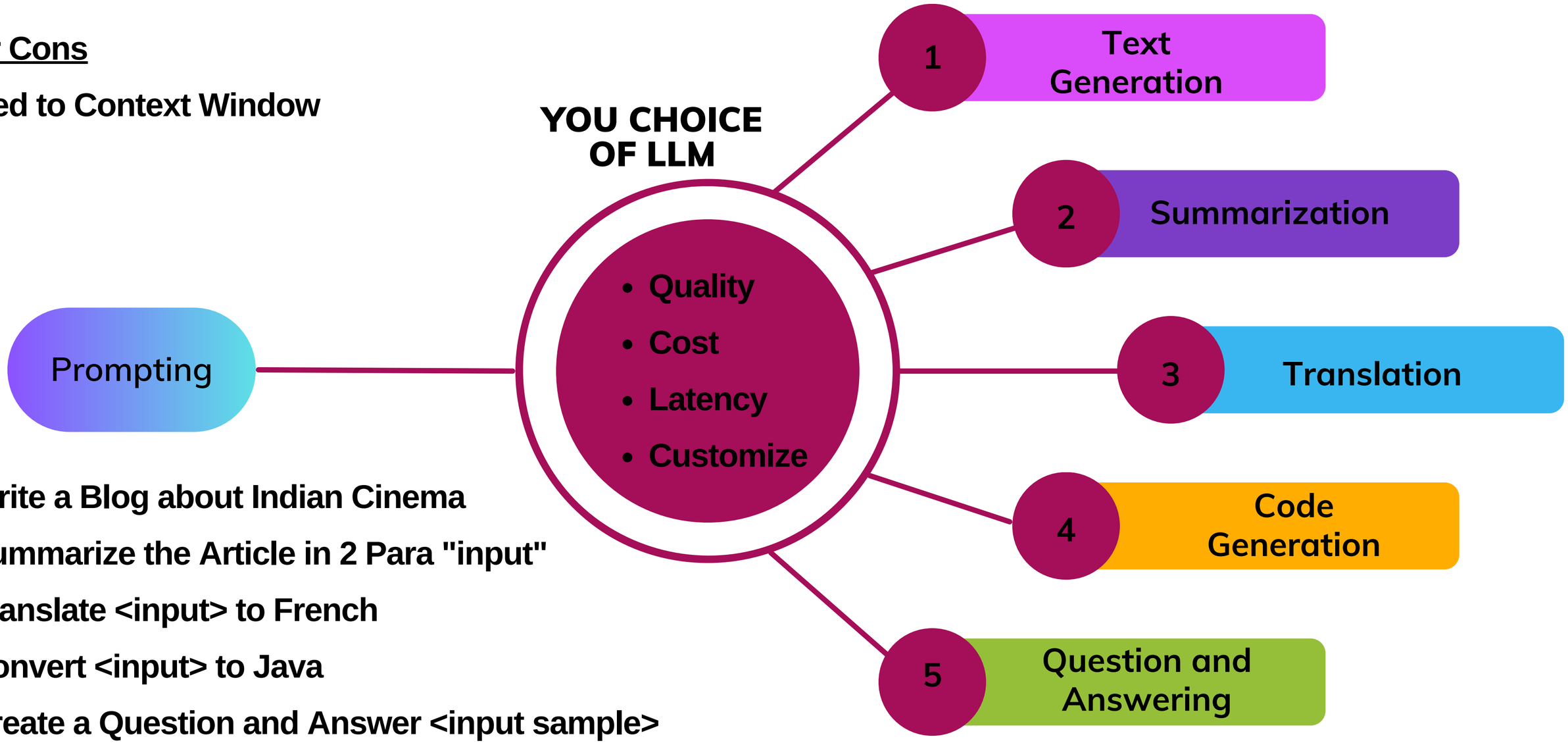


Sagemaker Foundation Model - Playground Console

LLM - Prompting

Major Cons

Limited to Context Window



1. **Write a Blog about Indian Cinema**
2. **Summarize the Article in 2 Para "input"**
3. **Translate <input> to French**
4. **Convert <input> to Java**
5. **Create a Question and Answer <input sample>**

LLM - Prompting

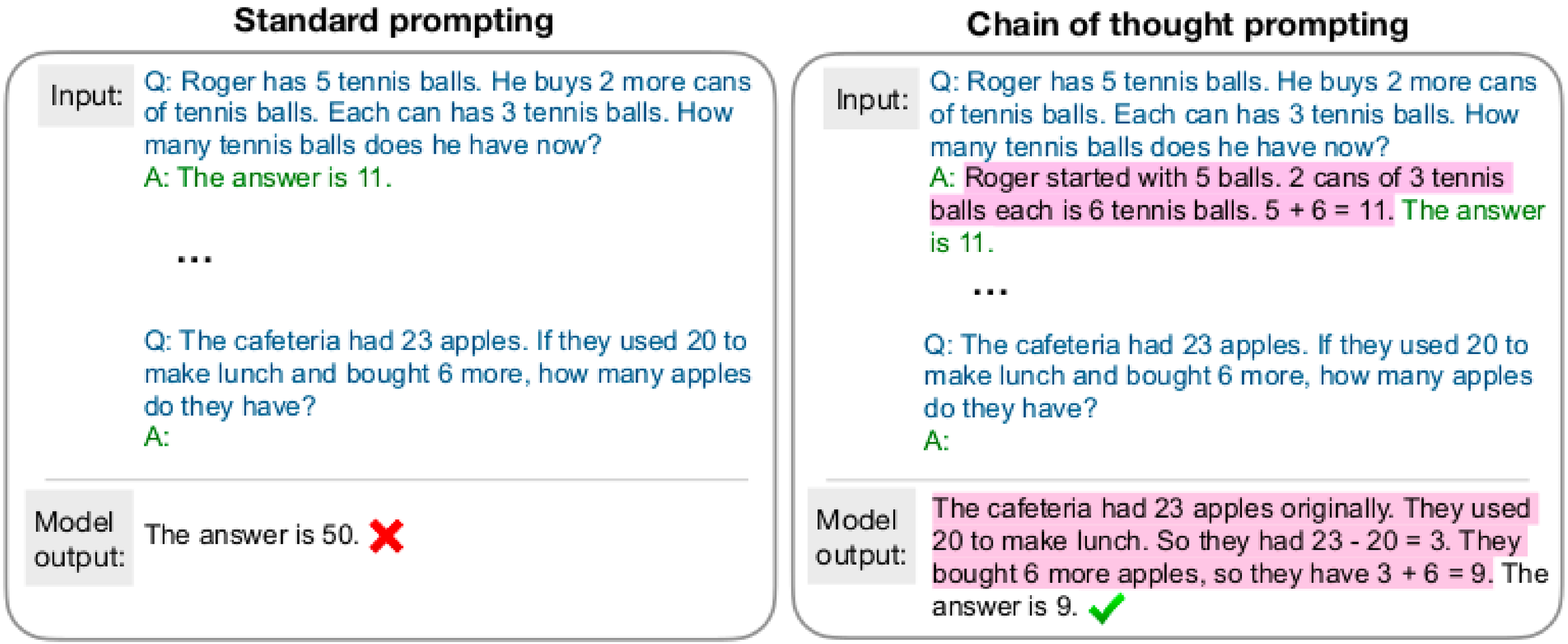
Zero Shot - Task described, but demonstrations not given

```
1  Translate English to French:        ← task description
2  cheese =>                           ← prompt
```

Few-shot: Task described and random demonstrations provided

```
1  Translate English to French:        ← task description
2  sea otter => loutre de mer          ← example
3  cheese =>                           ← prompt
```

Chain of Thoughts Prompting



Connect with Large Language via API - Persona - Developers - Prompting

```python
# Generate the content
import cohere
co = cohere.Client(api_key) # This is your trial API key
response = co.generate(
  model='command',
  prompt='Write a LinkedIn post about starting a career in tech:',
  max_tokens=300,
  temperature=0.9,
  k=0,
  stop_sequences=[],
  return_likelihoods='NONE')
print('Prediction: {}'.format(response.generations[0].text))
```

Sagemaker Foundation Models - Connect via API

LLM - Langchain

LLMs and Prompts

This includes prompt management, prompt optimization, a generic interface for all LLMs, and common utilities for working with LLMs.

Chains

Chains go beyond a single LLM call and involve sequences of calls (LLM + utility). LangChain provides a standard interface for chains, lots of integrations with other tools, and end-to-end chains for common applications.

Data Augmented Generation

It involves specific types of chains that first interact with an external data source to fetch data for use in the generation step. E.g) Summarization and QA

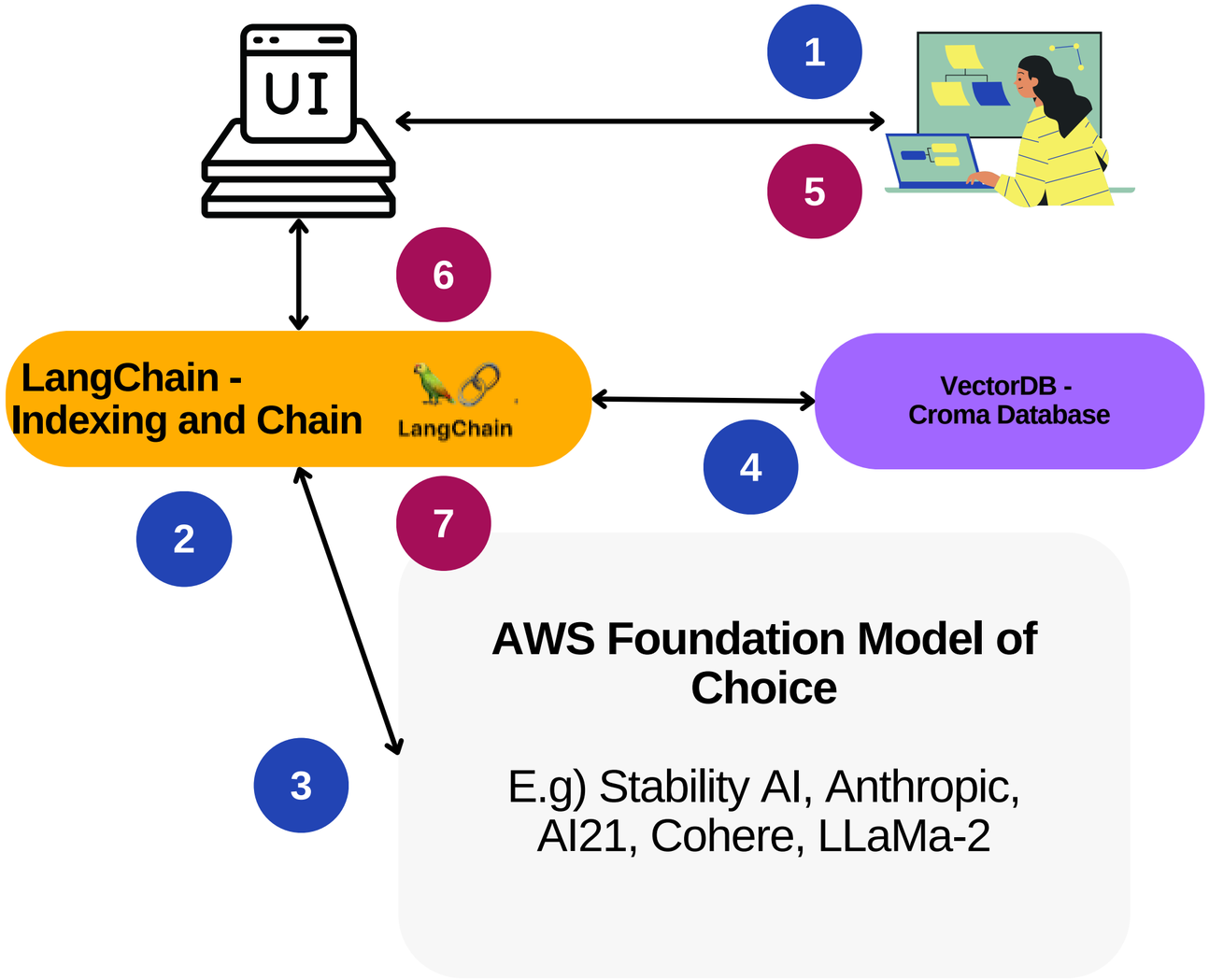
Agents

It involve an LLM making decisions about which Actions to take, taking that Action, seeing an Observation, and repeating that until done. It provides a standard interface for agents, a selection of agents to choose from, and examples of end-to-end agents

Memory

Memory refers to persisting state between calls of a chain/agent. LangChain provides a standard interface for memory, a collection of memory implementations, and examples of chains/agents that use memory.

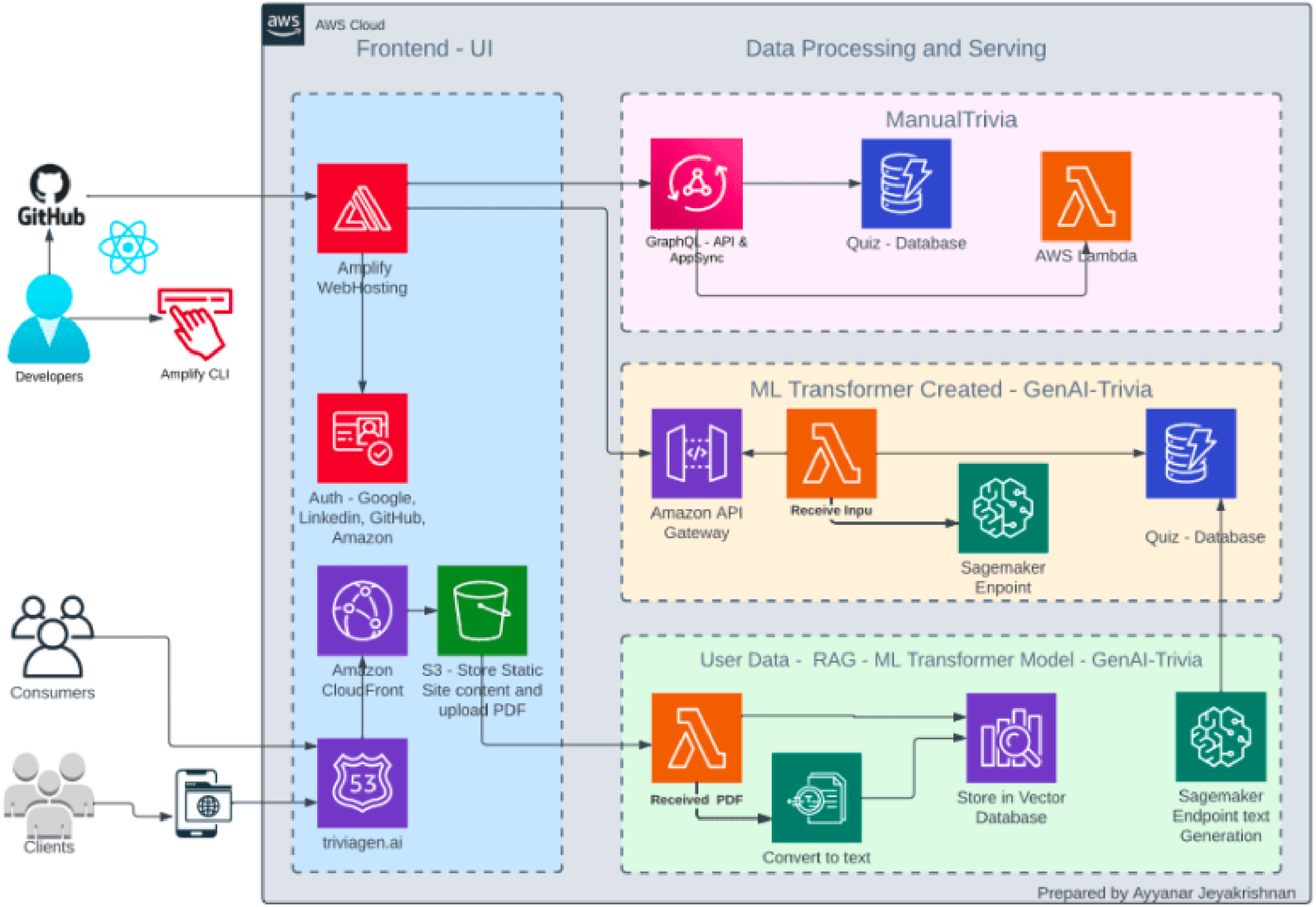
Small Application - Student Study App - P



1. Student upload PDF file via UI
2. Application built on Langchain read the PDF file as chunk and
3. Use Embed Model from Cohere to covert word to Vector
4. Upload Vector to Vector DB - Croma Database

5. Student Provide the Prompt from UI
6. Langchain use the Prompt Template and read the Input and forward the request to LLM and get the context / Semantic search from Vector DB
7. Use the Cohere API and return the result

Sample Usecase - Fun Project



https://dev.to/jayyanar/triviagenai-m98

Feedback

bit.ly/feedback-awscommunity

Thank you!

Ayyanar Jeyakrishnan

QUESTIONS

WELCOME