Applied Deep Learning

5350 Watsonx Lab Quiz

portilla@gmail.com

Jesus.portilla@Colorado.edu

https://github.com/iportilla/watsonx

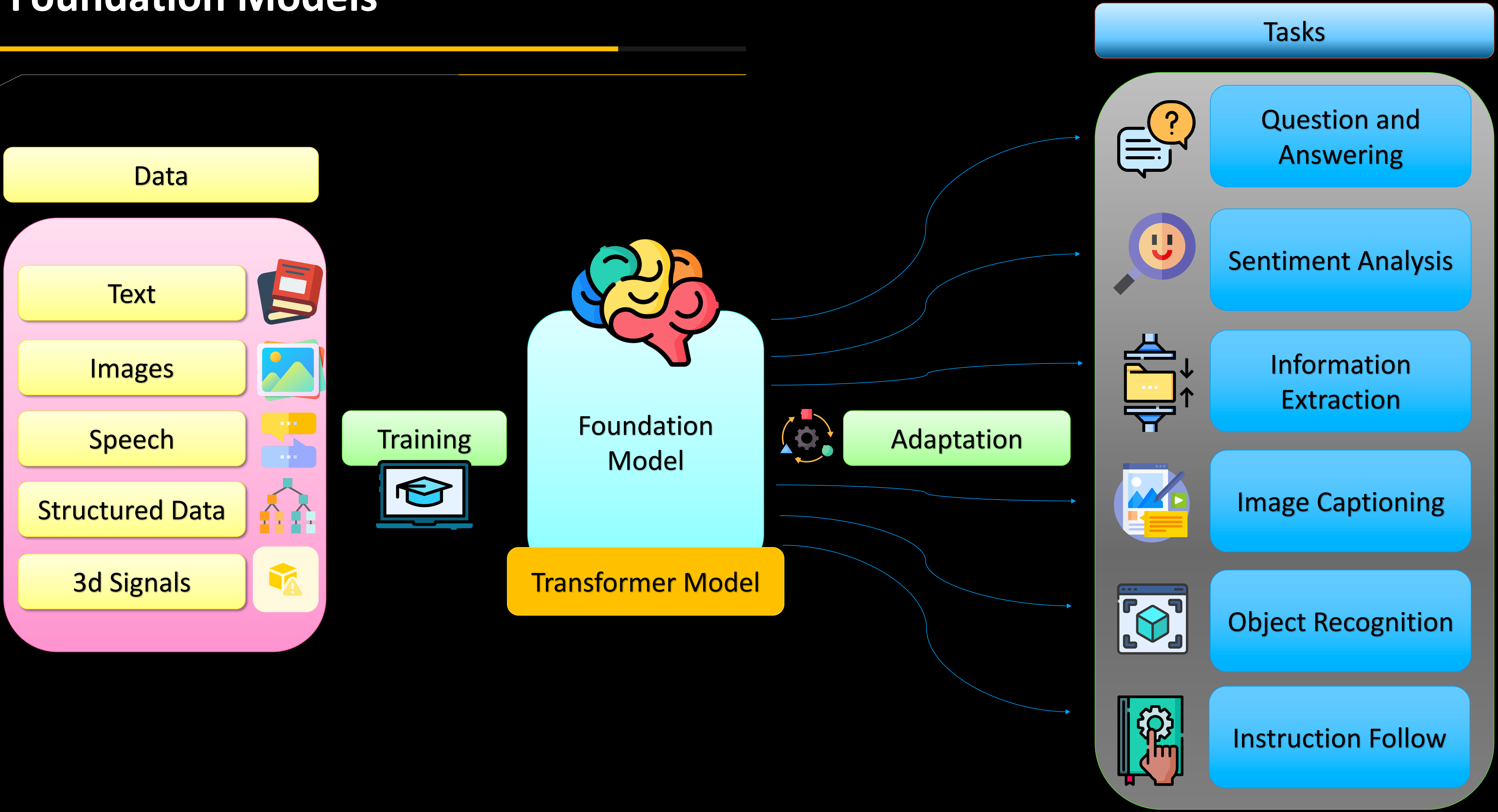
5350 Watsonx Lab Quiz

Cloud.ibm.com

Dataplatform.cloud.ibm.com

https://github.com/iportilla/watsonx

Foundation Models



Data

- Text
- Images
- Speech
- Structured Data
- 3d Signals

Training

Foundation Model

Transformer Model

Adaptation

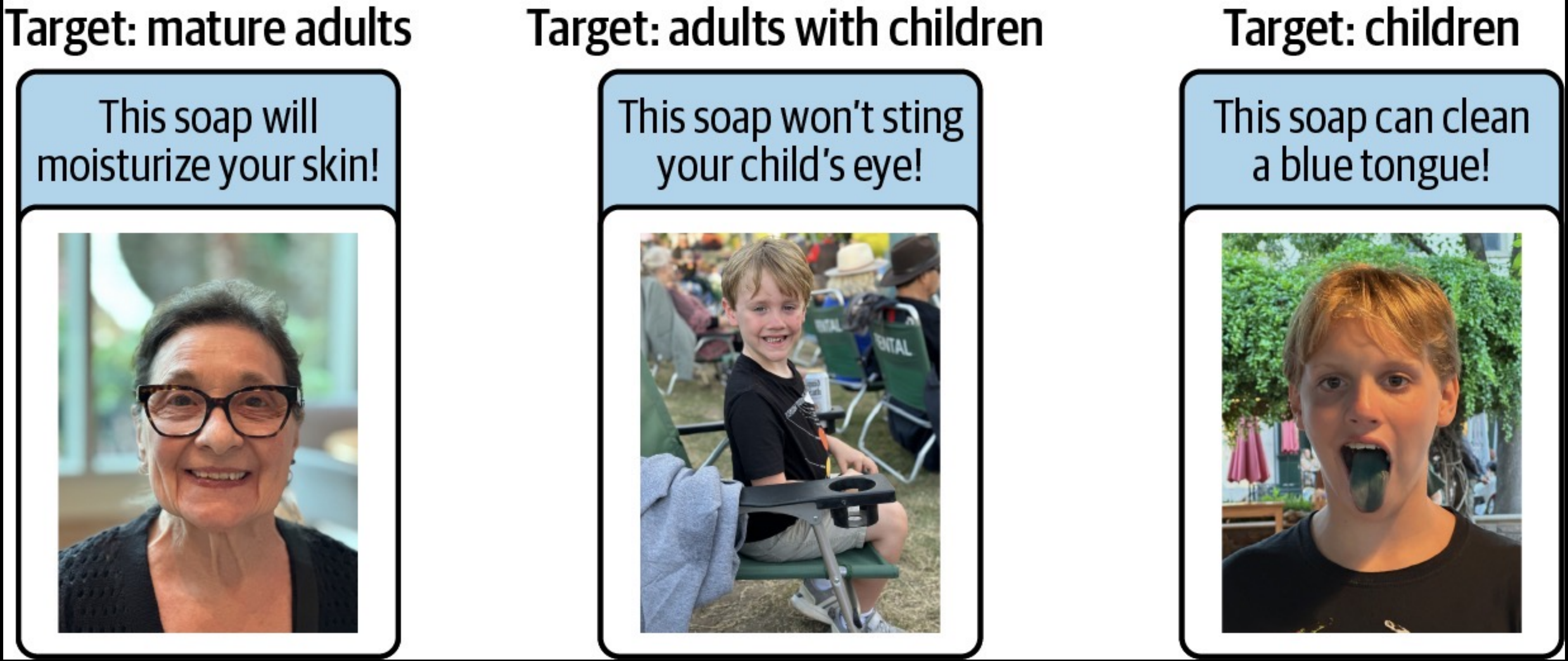
Tasks

- Question and Answering
- Sentiment Analysis
- Information Extraction
- Image Captioning
- Object Recognition
- Instruction Follow

Use cases & tasks

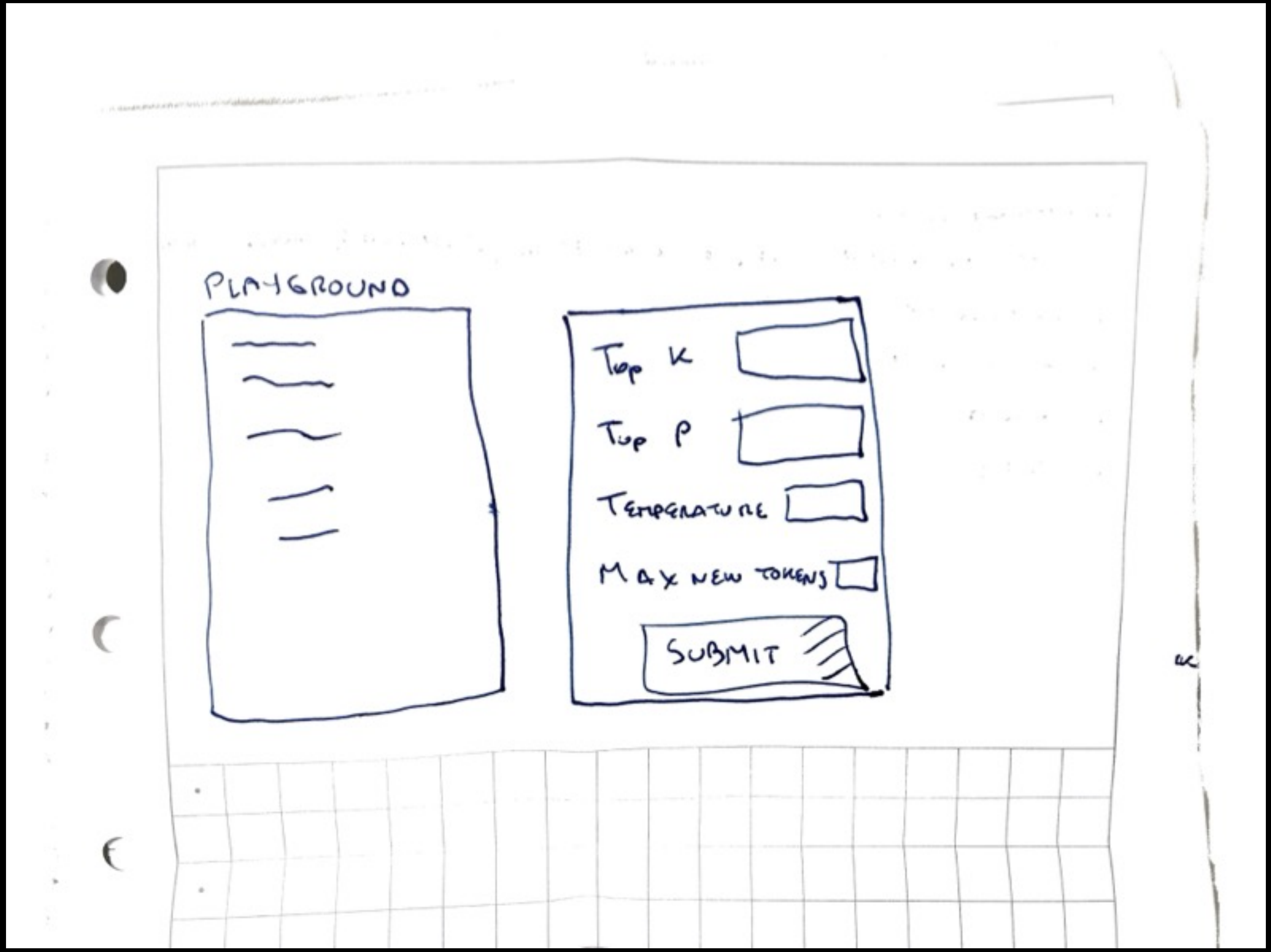
- Text summarization
- Rewriting
- Information extraction
- Q&A and Visual Q&A
- Detecting toxic / harmful content
- Classification & content moderation
- Conversational interface (chatbot)
- Language translation
- Source code generation
- Reasoning
- Mask personally identifiable information (PII)
- Personalized marketing and ads

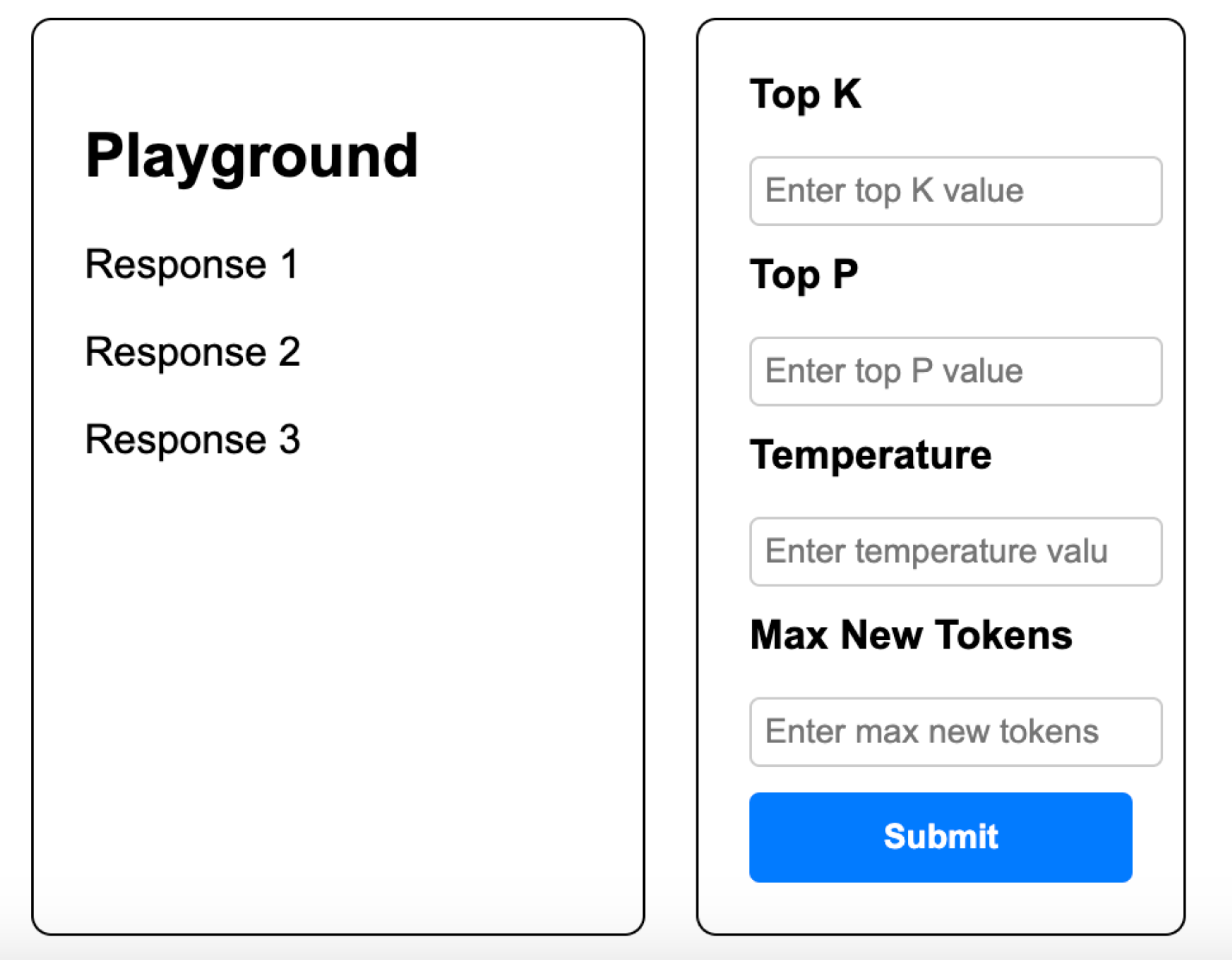
Use cases & tasks


Target: mature adults
This soap will moisturize your skin!
Target: adults with children
This soap won't sting your child's eye!
Target: children
This soap can clean a blue tongue!

Use cases & tasks

you are an expert UI designer, I will provide a sketch of a web page, write the html and javascript based on this picture





Playground

Response 1

Response 2

Response 3

Top K

Enter top K value

Top P

Enter top P value

Temperature

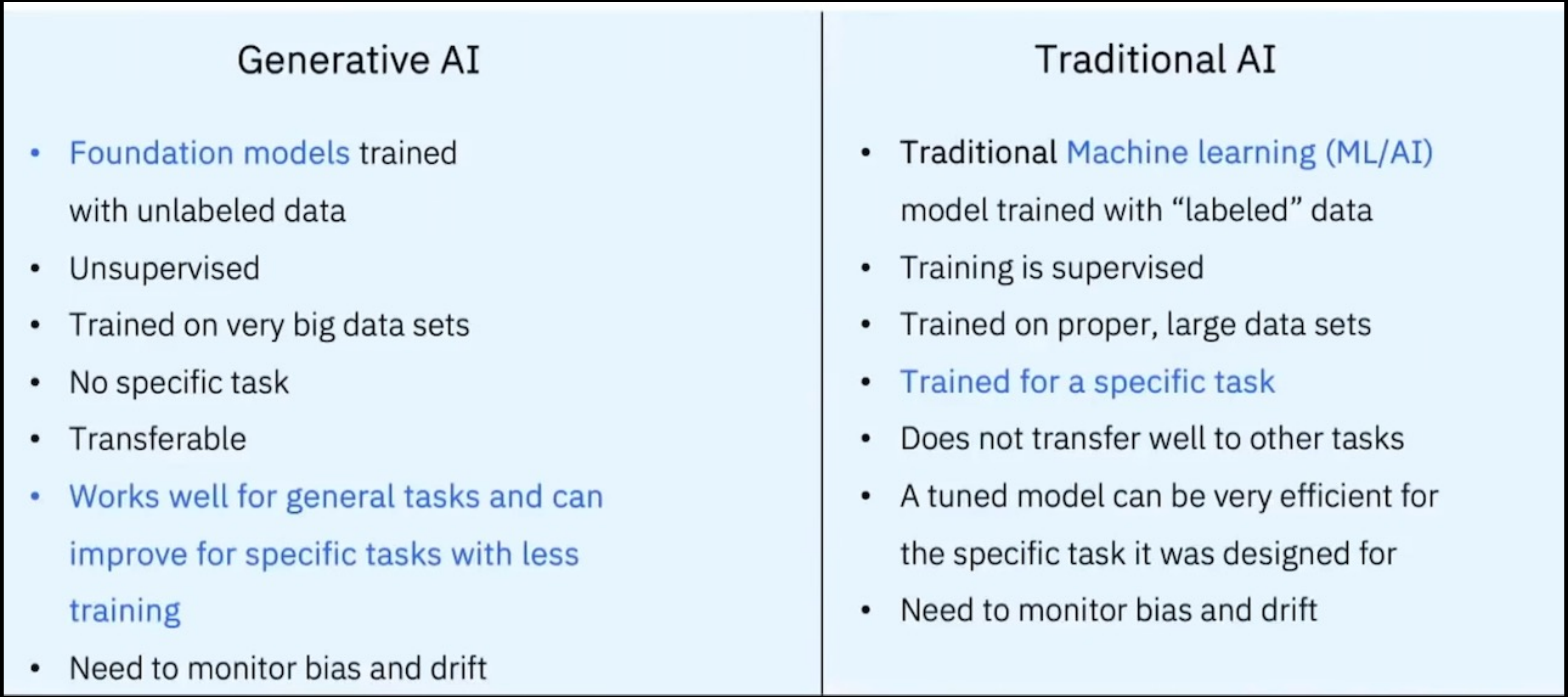
Enter temperature valu

Max New Tokens

Enter max new tokens

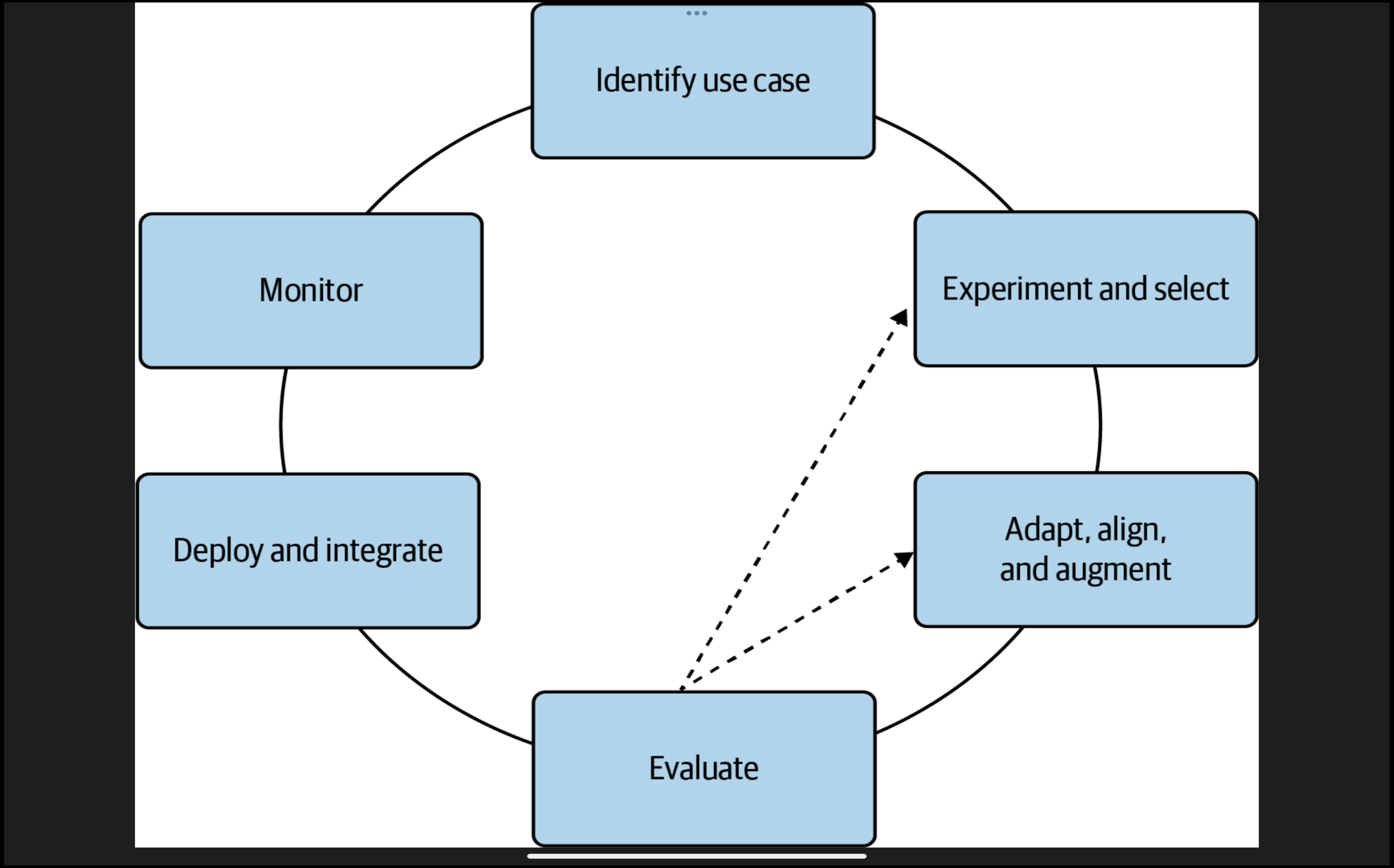
- Intros
- ChatGPT 101
- Business Examples
- Exercise

Generative AI and Traditional AI



Generative AI	Traditional AI
Foundation models trained with unlabeled data	Traditional Machine learning (ML/AI) model trained with “labeled” data
Unsupervised	Training is supervised
Trained on very big data sets	Trained on proper, large data sets
No specific task	Trained for a specific task
Transferable	Does not transfer well to other tasks
Works well for general tasks and can improve for specific tasks with less training	A tuned model can be very efficient for the specific task it was designed for
Need to monitor bias and drift	Need to monitor bias and drift

GenerativeAI Project Lifecyle



https://github.com/generative-ai-on-aws/generative-ai-on-aws

How to select the best GenAI Model

How to select the best GenAI Model

1. Understand biz case

2. List models available

3. Identify each model +/-

4. Evaluate model characteristics

5. Run some model tests

6. Choose the best option

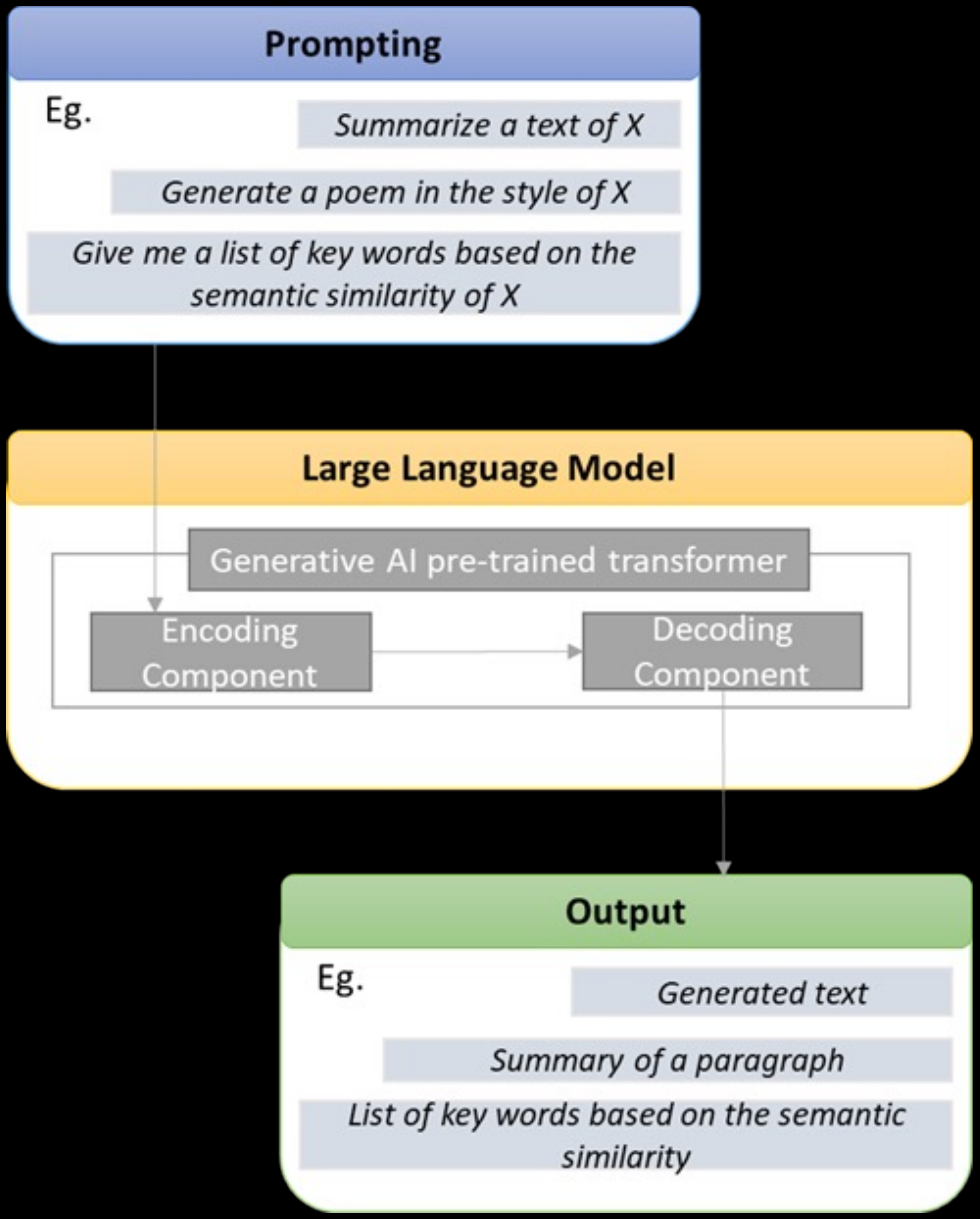
https://youtu.be/pePAAGfh-IU?si=8FqgKIj6Oxy6FjUc

IBM watsonx.ai is a studio of integrated tools for working with generative AI capabilities that are powered by foundation models and for building machine learning models.

IBM watsonx.ai provides a secure and collaborative environment where you can access your organization's trusted data, automate AI processes, and deliver AI in your applications.

https://video.ibm.com/channel/23952663/video/wx-ai-overview

GenAI



Prompting

Eg.

Summarize a text of X

Generate a poem in the style of X

Give me a list of key words based on the semantic similarity of X

Large Language Model

Generative AI pre-trained transformer

Encoding Component

Decoding Component

Output

Eg.

Generated text

Summary of a paragraph

List of key words based on the semantic similarity

watsonx

Summarizaton

- Meeting transcript summary
- Earnings call summary

Classification

- Scenario classification
- Sentiment classification

Generation

- Marketing email generation
- Thank you note generation

Extraction

- Named entity extraction
- Fact extraction

Question and Answering

- Questions about an article
- Finance Q&A

Code

- Code generation
- Code translation

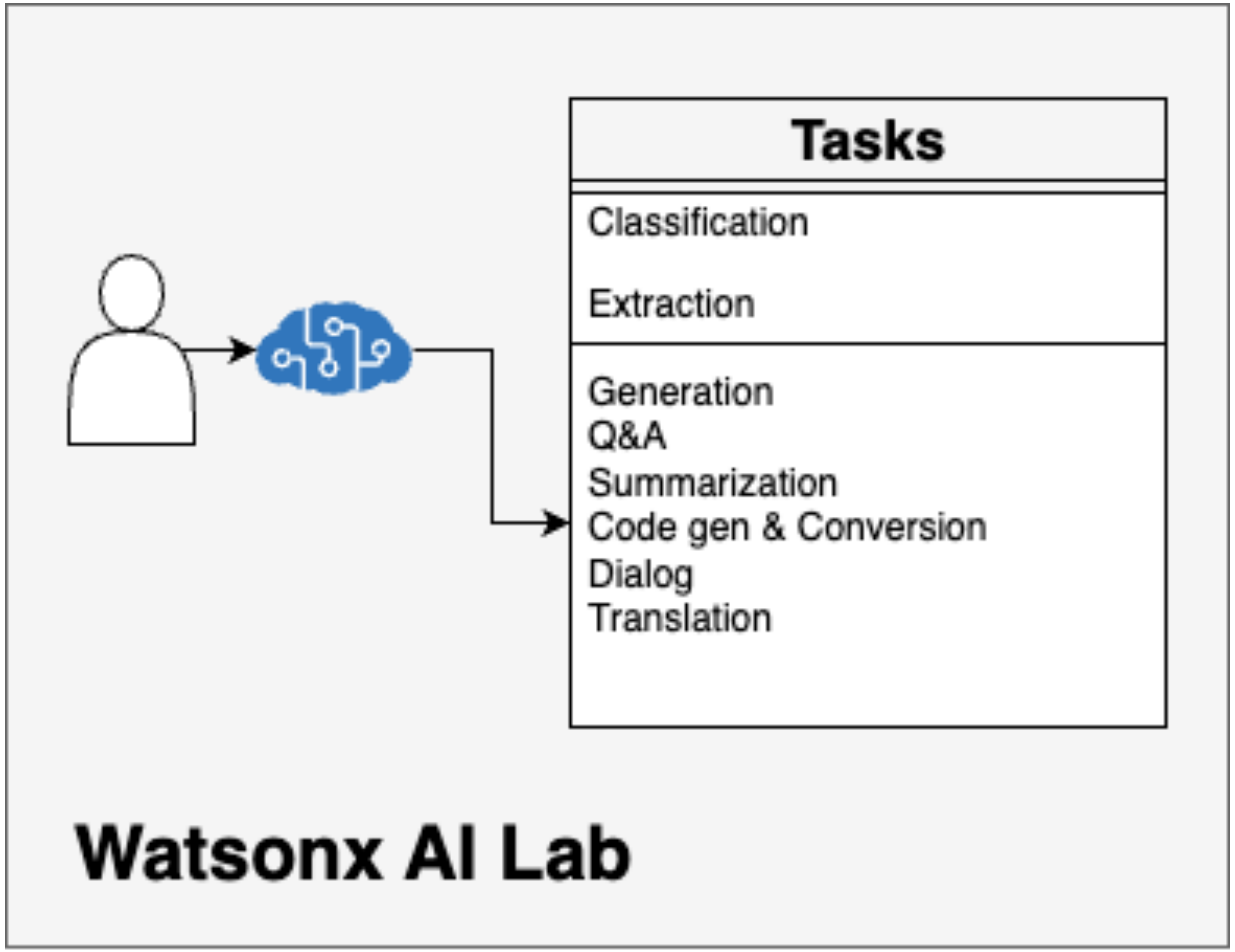
Class exercise

IBM watsonx studio

https://dataplatform.cloud.ibm.com/

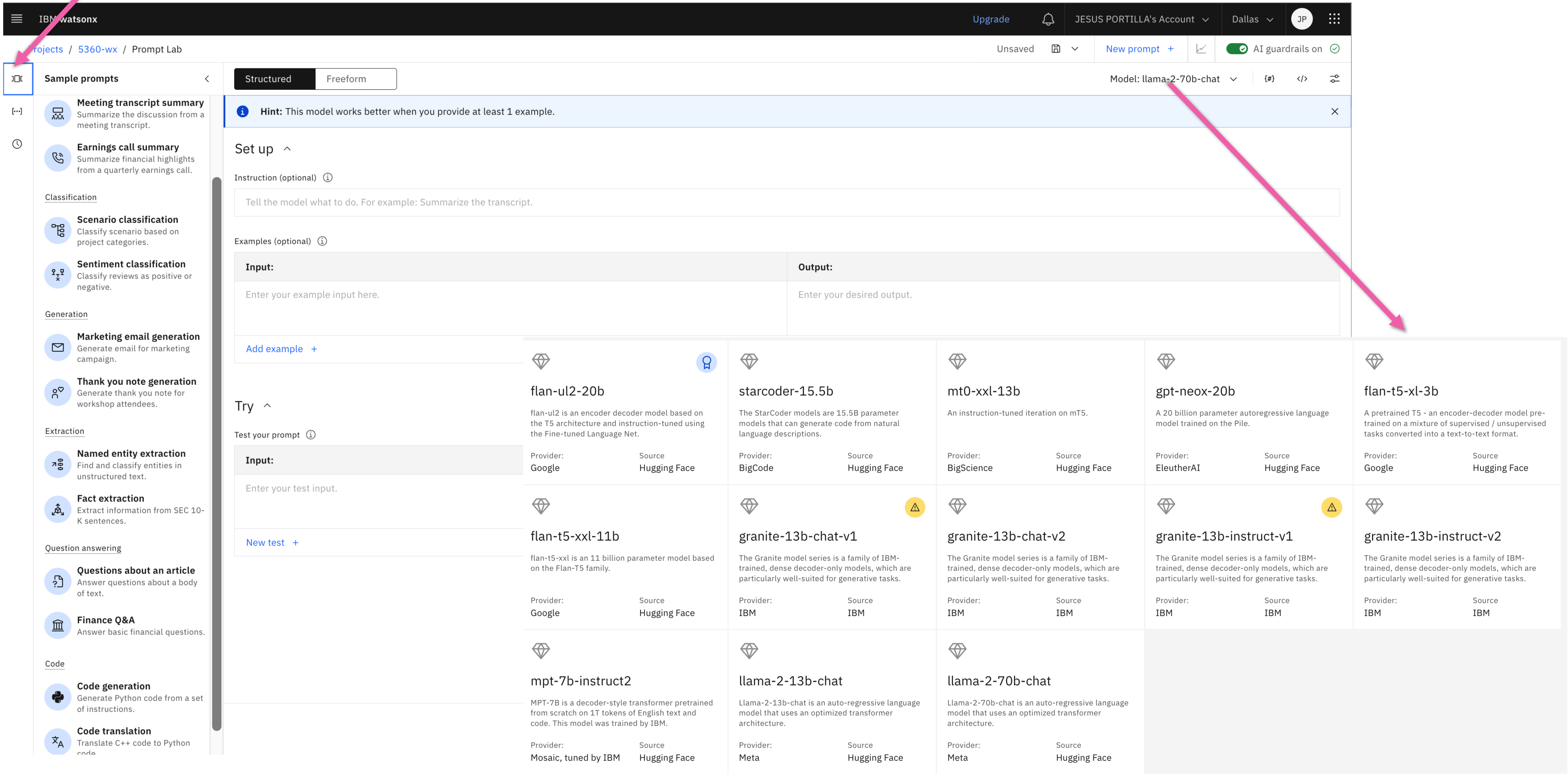
Class exercise

IBM watsonx studio
@
dataplatform.ibm.com



Class exercise

IBM watsonx studio



Meeting transcript summary
Summarize the discussion from a meeting transcript.

Earnings call summary
Summarize financial highlights from a quarterly earnings call.

Classification

Scenario classification
Classify scenario based on project categories.

Sentiment classification
Classify reviews as positive or negative.

Generation

Marketing email generation
Generate email for marketing campaign.

Thank you note generation
Generate thank you note for workshop attendees.

Extraction

Named entity extraction
Find and classify entities in unstructured text.

Fact extraction
Extract information from SEC 10-K sentences.

Question answering

Questions about an article
Answer questions about a body of text.

Finance Q&A
Answer basic financial questions.

Code

Code generation
Generate Python code from a set of instructions.

Code translation
Translate C++ code to Python code

Hint: This model works better when you provide at least 1 example.

Set up

Instruction (optional)

Tell the model what to do. For example: Summarize the transcript.

Examples (optional)

Input:	Output:
Enter your example input here.	Enter your desired output.

Add example +

Try

Test your prompt

Input:

Enter your test input.

New test +

Model	Description	Provider	Source
flan-ul2-20b	flan-ul2 is an encoder decoder model based on the T5 architecture and instruction-tuned using the Fine-tuned Language Net.	Google	Hugging Face
starcoder-15.5b	The StarCoder models are 15.5B parameter models that can generate code from natural language descriptions.	BigCode	Hugging Face
mt0-xxl-13b	An instruction-tuned iteration on mT5.	BigScience	Hugging Face
gpt-neox-20b	A 20 billion parameter autoregressive language model trained on the Pile.	EleutherAI	Hugging Face
flan-t5-xl-3b	A pretrained T5 - an encoder-decoder model pre-trained on a mixture of supervised / unsupervised tasks converted into a text-to-text format.	Google	Hugging Face
flan-t5-xxl-11b	flan-t5-xxl is an 11 billion parameter model based on the Flan-T5 family.	Google	Hugging Face
granite-13b-chat-v1	The Granite model series is a family of IBM-trained, dense decoder-only models, which are particularly well-suited for generative tasks.	IBM	IBM
granite-13b-chat-v2	The Granite model series is a family of IBM-trained, dense decoder-only models, which are particularly well-suited for generative tasks.	IBM	IBM
granite-13b-instruct-v1	The Granite model series is a family of IBM-trained, dense decoder-only models, which are particularly well-suited for generative tasks.	IBM	IBM
granite-13b-instruct-v2	The Granite model series is a family of IBM-trained, dense decoder-only models, which are particularly well-suited for generative tasks.	IBM	IBM
mpt-7b-instruct2	MPT-7B is a decoder-style transformer pretrained from scratch on 1T tokens of English text and code. This model was trained by IBM.	Mosaic, tuned by IBM	Hugging Face
llama-2-13b-chat	Llama-2-13b-chat is an auto-regressive language model that uses an optimized transformer architecture.	Meta	Hugging Face
llama-2-70b-chat	Llama-2-70b-chat is an auto-regressive language model that uses an optimized transformer architecture.	Meta	Hugging Face

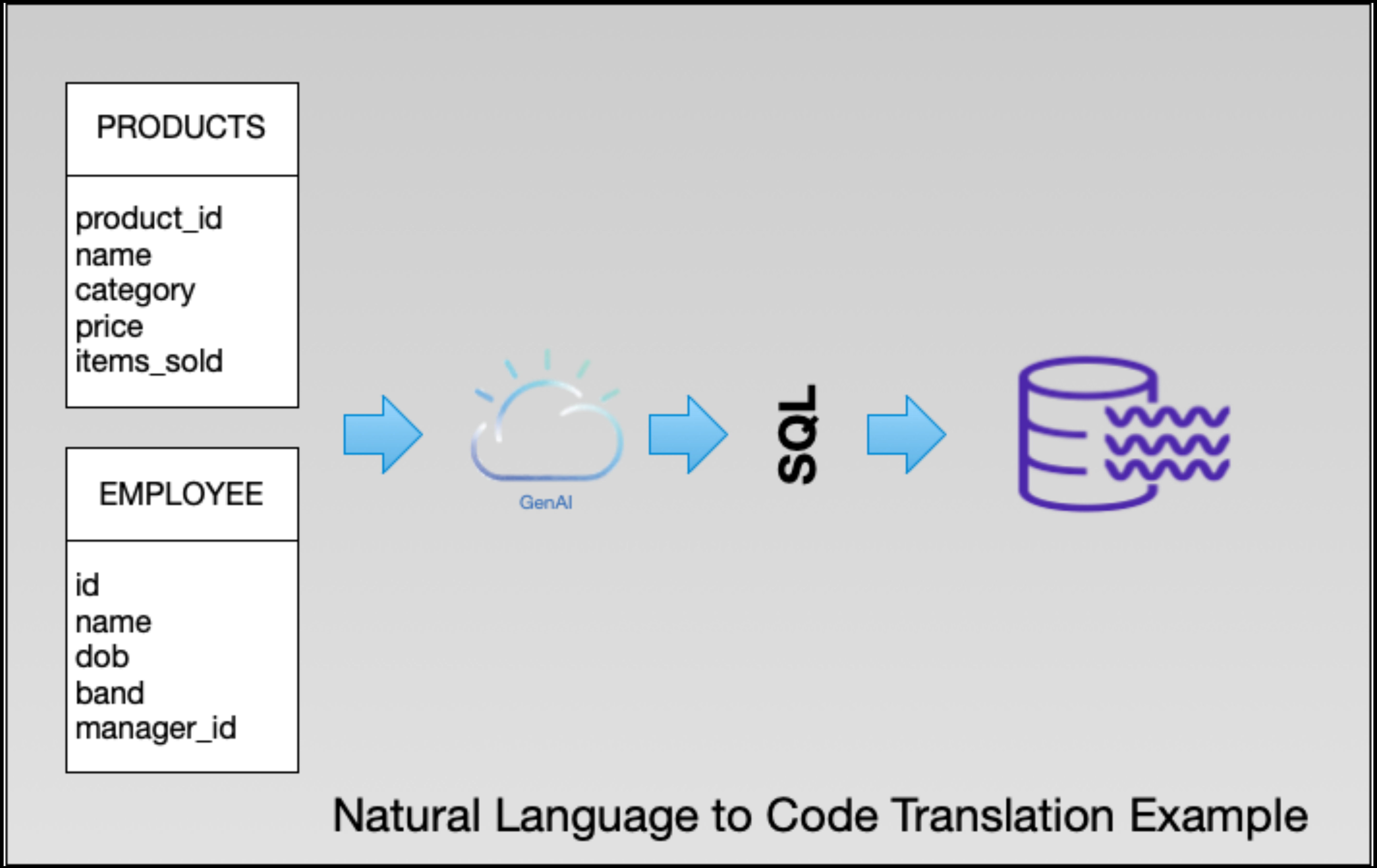
Lab 2

English & Code Translation

Query top five products based on the price and items sold

Query the count of employees in band L6 and with manager ID as 23079

Write SQL Query given the table name is {Table} and columns are {Columns} for the question : {question}.



Natural Language to Code Translation Example

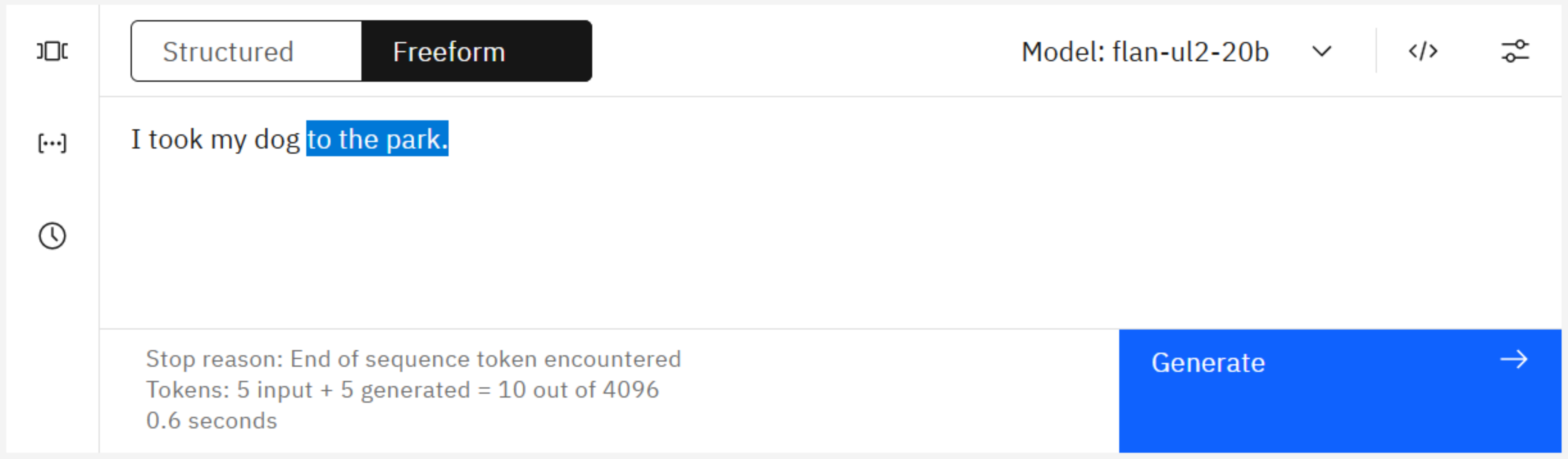
Prompt Tips

https://dataplatform.cloud.ibm.com/docs/content/wsj/analyze-data/fm-prompt-tips.html?context=wx

Tip 1: Remember that everything is text completion

Your *prompt* is the text you submit for processing by a foundation model.
For most models, simply asking a question or typing an instruction won't yield the best results. That's because the model isn't *answering* your prompt, the model is *appending text to it*.

This image demonstrates prompt text and generated output:
- Prompt text: "I took my dog "
- Generated output: "to the park."



I took my dog to the park.

Stop reason: End of sequence token encountered
Tokens: 5 input + 5 generated = 10 out of 4096
0.6 seconds

Tip 2: Include all the needed prompt components

Effective prompts usually have one or more of the following components: instruction, context, examples, and cue.

Instruction

An instruction is an imperative statement that tells the model what to do. For example, if you want the model to list ideas for a dog-walking business, your instruction could be: "List ideas for starting a dog-walking business:”

Context

Including background or contextual information in your prompt can nudge the model output in the right direction. Specifically, words that appear in your prompt text are more likely to be included in the generated output.

Examples

To indicate the format or shape that you want the model response to be, include one or more pairs of example input and corresponding output that illustrate the pattern you want the generated text to follow.
Including one example in your prompt is called *one-shot prompting*, including two or more examples in your prompt is called *few-shot* prompting, and when your prompt has no examples, that's called *zero-shot* prompting.
When you prompt models that have been tuned, you might not need to include examples.

Cue

A cue is text at the end of the prompt that is likely to start the generated output on the right path. Remember, as much as it seems like the model is *responding to your prompt*, the model is really *appending text to your prompt* or *continuing your prompt*.

Tip 3: Include descriptive details

The more guidance, the better. Experiment with including descriptive phrases related to aspects of your ideal result: content, style, and length. Including these details in your prompt can cause a more creative or more complete result to be generated.

For example, you can improve the sample instruction by adding descriptive information as follows:

- Original: "List ideas for starting a dog-walking business"
- Improved: "List ideas for starting a large, wildly successful dog-walking business”

Example:

Before
A prompt with a simple instruction:
List ideas for starting a dog-walking business.
The resulting foundation model output isn't great:
Take care of the dog and walk it in your neighborhood. You can create flyers and put them around the neighborhood, you can also put up ads on websites where dog lovers hang out.

After
A prompt with all of the recommended prompt components:

Tip 3: Include descriptive details

Instruction:
Based on the following industry research, suggest ideas for starting a large, wildly successful dog-walking business.

Context:
Industry research:

***

The most successful dog-walking businesses cater to owners' needs and desires while also providing great care to the dogs. For example, owners want flexible hours, a shuttle to pick up and drop off dogs at home, and personalized services, such as custom meal and exercise plans. Consider too how social media has permeated our lives. Web-enabled interaction provide images and video that owners will love to share online, which is great advertising for the business.

***

Tip 3: Include descriptive details

Example

Ideas for starting a lemonade business:

- Set up a lemonade stand
- Partner with a restaurant
- Get a celebrity to endorse the lemonade

Cue:

Ideas for starting a large, wildly successful dog-walking business:

RESPONSE:

- *Start by doing some basic market research. Interview dog owners in your area and ask them what they would want from their dog walker.*
- *Offer a free sample or trial period - like several weeks - so you can see if they are really interested.*
- *Give personalized service. Offer to train the owner's dog to get in a proper sit or down for the dog walker. Train the human too!*
- *Be friendly and outgoing. Social media is huge and people love to post photos of themselves with their pets.*
- *Use technology to help keep costs low, such as using web-enabled interactions. For example, you could offer a coupon on a popular pet food store website.*

Prompting Principles

Principle 1

Write clear and specific instructions

Tactics:

- *Tactic 1 – Use Delimiters to clearly indicate distinct parts of the input*
- *Tactic 2 – Ak for structured output*
- *Tactic 3 – Ask the model to check whether conditions are satisfied*
- *Tactic 4 – Try ‘few—shot’ prompting*

Prompting Principles

Principle 2

Give the model time to think

Tactics:

- *Tactic 1 – Specify the steps require to complete a task*
- *Tactic 2 – Instruct the model to work out its own solution before rushing to conclusion*
- *Tactic 3 – Watch out for hallucinations*

Prompt Techniques Lab