



AWS BUILDERS KOREA PROGRAM SPECIAL

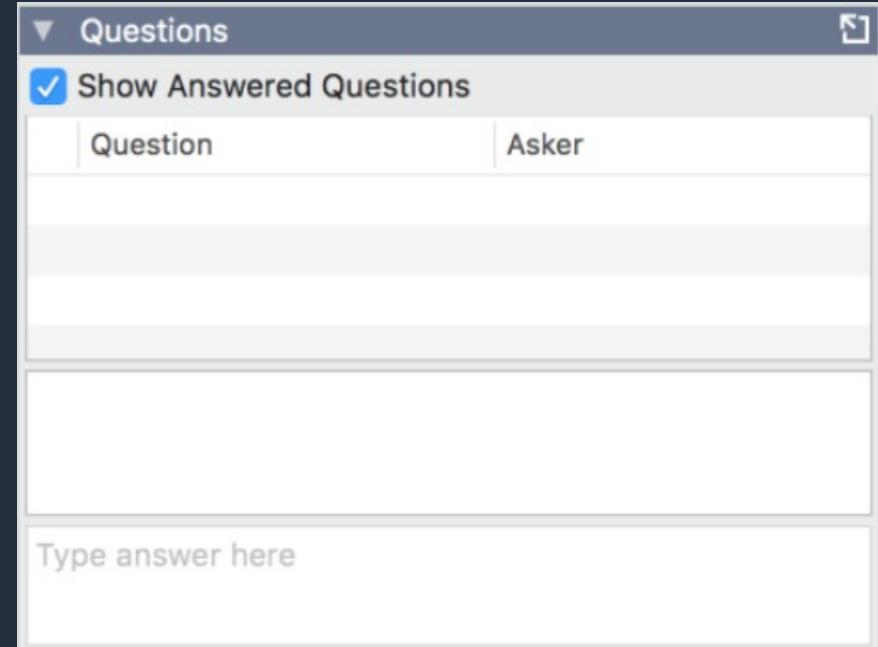
Foundation Model Customizing Technique

Hyo
Solutions Architect

강연 중 질문하는 방법

AWS Builders Korea Go to Webinar “**Questions (질문)**” 창에 자신이 질문한 내역이 표시됩니다. 본인만 답변을 받고 싶으실 경우 (비공개)라고 하고 질문해 주시면 됩니다.

질문 주신 사항에 대해서는 질문창을 통해 답변을 드립니다.



고지 사항 (Disclaimer)

본 컨텐츠는 고객의 편의를 위해 AWS 서비스 설명을 위해 온라인 세미나용으로 별도로 제작, 제공된 것입니다. 만약 AWS 사이트와 컨텐츠 상에서 차이나 불일치가 있을 경우, AWS 사이트(aws.amazon.com)가 우선합니다. 또한 AWS 사이트 상에서 한글 번역문과 영어 원문에 차이나 불일치가 있을 경우(번역의 지체로 인한 경우 등 포함), 영어 원문이 우선합니다.

AWS는 본 컨텐츠에 포함되거나 컨텐츠를 통하여 고객에게 제공된 일체의 정보, 콘텐츠, 자료, 제품(소프트웨어 포함) 또는 서비스를 이용함으로 인하여 발생하는 여하한 종류의 손해에 대하여 어떠한 책임도 지지 아니하며, 이는 직접 손해, 간접 손해, 부수적 손해, 징벌적 손해 및 결과적 손해를 포함하되 이에 한정되지 아니합니다.

실습 시작 전 준비 사항

AWS 계정으로 시작

1. 실습 전 계정을 꼭 신청해주세요 : <https://portal.aws.amazon.com/billing/signup#/start>
2. AWS 계정이 없으신 경우, 행사 참여 전에 미리 AWS 계정 생성 가이드를 확인하시고 AWS 계정을 생성해 주시길 바랍니다.

*AWS 계정 생성 가이드: <https://aws.amazon.com/ko/premiumsupport/knowledge-center/create-and-activate-aws-account/>

3. 검증된 호환성을 위하여 실습 시 사용할 웹 브라우저는 Mozilla Firefox 또는 Google Chrome Browser로 진행 부탁드립니다.

실습 마무리 및 설문 참여 방법

- 실습이 모두 끝난 후에는 **자원 삭제**를 잊지 마세요. 직접 준비하신 AWS 계정으로 실습을 진행하신 고객 분들의 경우, 가이드에 따라 자원 삭제를 진행하셔야 합니다. 또한, 기존에 사용하시던 자원이 있으신 고객 분들의 경우, **오늘 생성한 자원만 삭제하는 것에 주의 부탁드립니다.**
- **가이드:** (세션별 제공)
- 마지막으로 세션이 끝난 후, **GoToWebinar** 창을 종료하면 설문 조사 창이 나옵니다. 이때, **설문 조사를 진행해 주셔야 AWS 크레딧**(1인당 \$50 크레딧, 전체 세션당 1회 제공)을 제공받으실 수 있습니다.

AWS는 고객 피드백을 기반으로 의사 결정을 수행하며 이러한 피드백은 추후에 진행할 세션 방향을 결정합니다.

더 나은 세션을 위하여 여러분들의 소중한 의견을 부탁드립니다.

감사합니다.

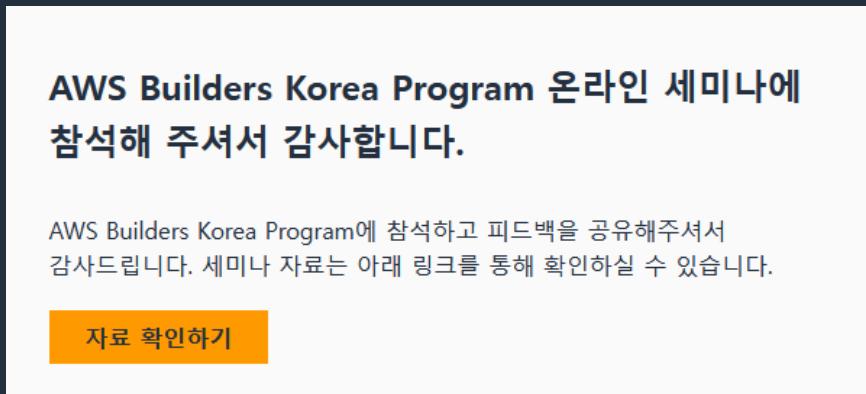
크레딧 안내

- AWS 계정으로 시작하실 경우, 금일 실습에서 발생하는 비용은 당월 과금이 되는 점 미리 확인 부탁 드립니다.
- 웨비나 종료 후 설문 조사에 참여해주신 분들께는 AWS 크레딧 바우처 (1인당 \$50 USD 크레딧, 전체 세션당 1회 제공)를 드립니다.
- 해당 AWS 크레딧은 등록하신 이메일 계정으로 행사 종료 후 1개월 내 발송 드릴 예정이며, 전달 받은 AWS 크레딧은 바로 사용 가능합니다.

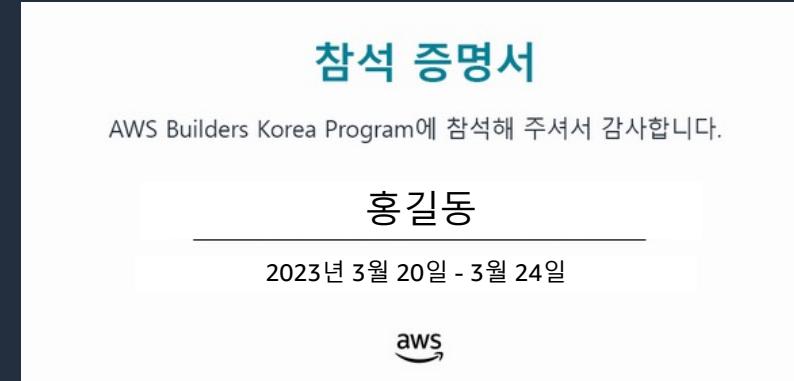
감사 메일 & 참석 증명서

- AWS Builders Korea 세션에 참석해 주신 분들께 행사 종료 후 1개월 내 감사메일과 참석 증명서가 순차 발송됩니다.
- 등록 진행 후 참석하지 않으실 경우 별도 메일 및 증명서는 발급되지 않습니다.

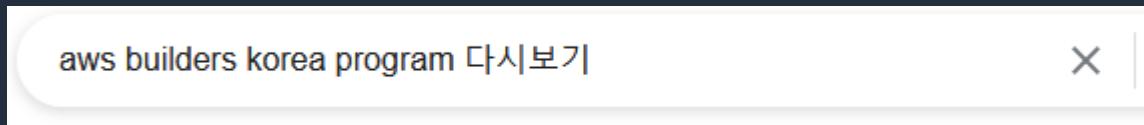
감사 메일 예시



참석 증명서 예시

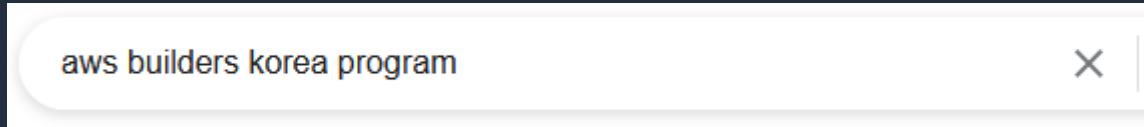


강연 다시보기



<https://kr-resources.awscloud.com/aws-builders-korea-program>

AWS Builders Korea 프로그램 정보



<https://aws.amazon.com/ko/events/seminars/aws-builders/>

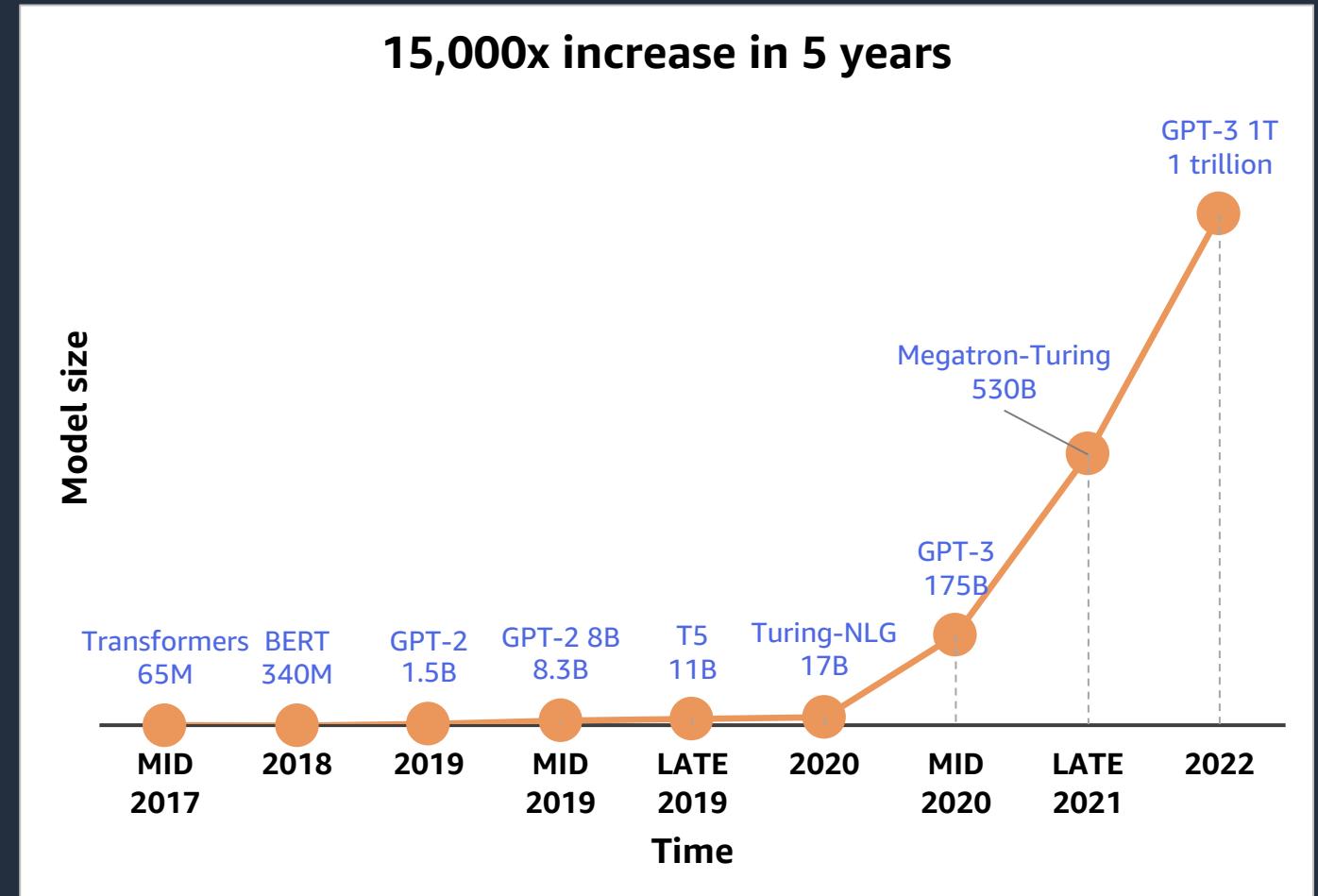
Foundation Model Customizing Technique

Size of large NLP models is increasing

기존 ML에서 딥 러닝으로 전환하면서
모델이 더욱 복잡해짐

모델이 클수록 더 잘 일반화된다는 사실
이 밝혀지면서 최신 딥 러닝 모델의 규
모가 점점 더 커지고 있음

더 나은 검색, 제품 추천, 챗봇, 이미지
생성, 시각적 질문/답변, 감정 인식 등
다양한 사례에 활용되며 실험 ing...



FM Use cases

Audio

- Text-to-Speech
- Automatic Speech Recognition
- Audio-to-Audio
- Audio Classification
- Voice Activity Detection

Tabular

- Tabular Classification
- Tabular Regression

Reinforcement Learning

- Reinforcement Learning
- Robotics

Multimodal

- Feature Extraction
- Text-to-Image
- Image-to-Text
- Visual Question Answering
- Document Question Answering
- Graph Machine Learning

Computer Vision

- Depth Estimation
- Image Classification
- Object Detection
- Image Segmentation
- Image-to-Image
- Unconditional Image Generation
- Video Classification
- Zero-Shot Image Classification

Natural Language Processing

- Text Classification
- Token Classification
- Table Question Answering
- Question Answering
- Zero-Shot Classification
- Translation
- Summarization
- Conversational
- Text Generation
- Text2Text Generation
- Fill-Mask
- Sentence Similarity

2023 LLM

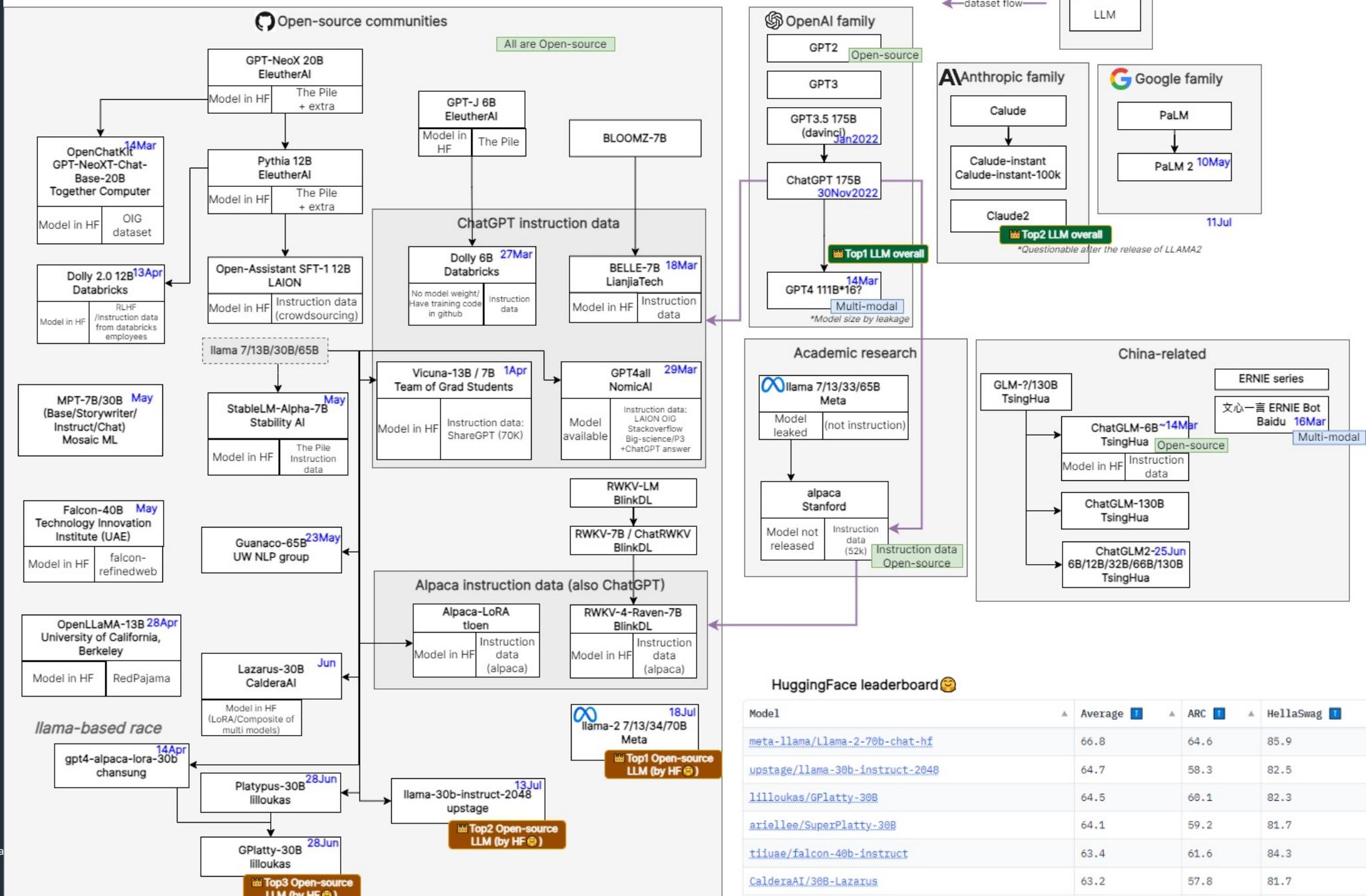
May-Jul 2023 - Recent Instruction/Chat-Based Models and their parents

As of 20230719

github/michaelthwan

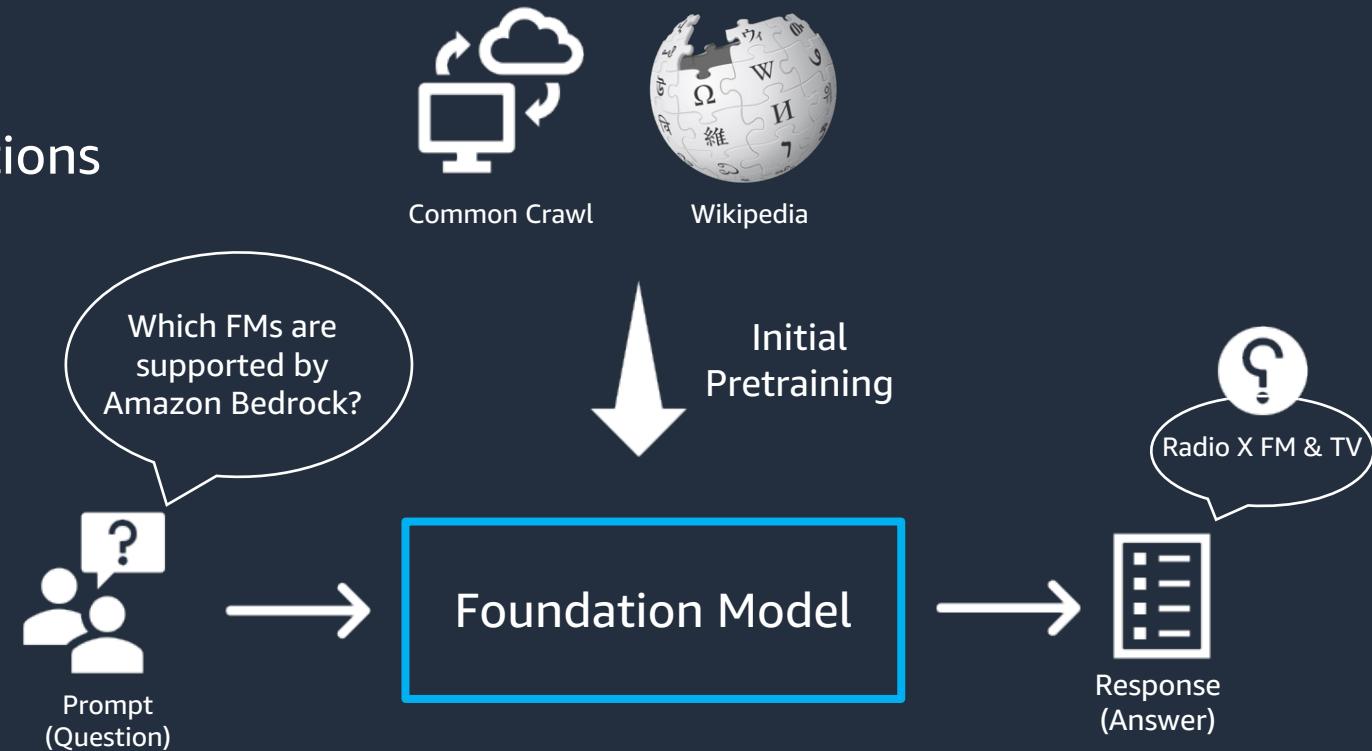
Rough release date

←Inheritance—
↔dataset flow—



Why customize a foundation model?

- Improving the performance/quality
- Specific Task
- Closed-domain knowledge
- Current Knowledge
- Reduce likelihood of hallucinations
- Out of Memory



Emerging LLM customisation patterns

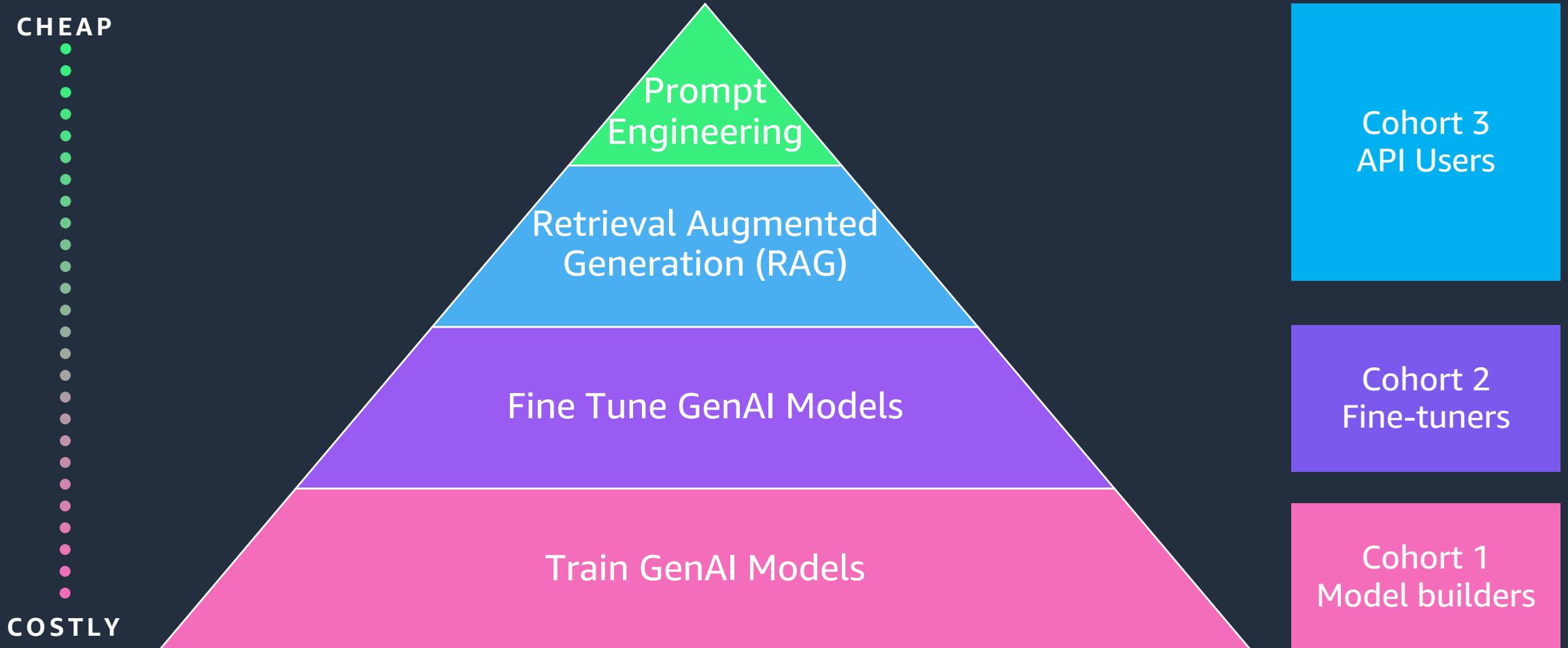
Prompt Engineering (with context)

Retrieval Augmented Generation (RAG)

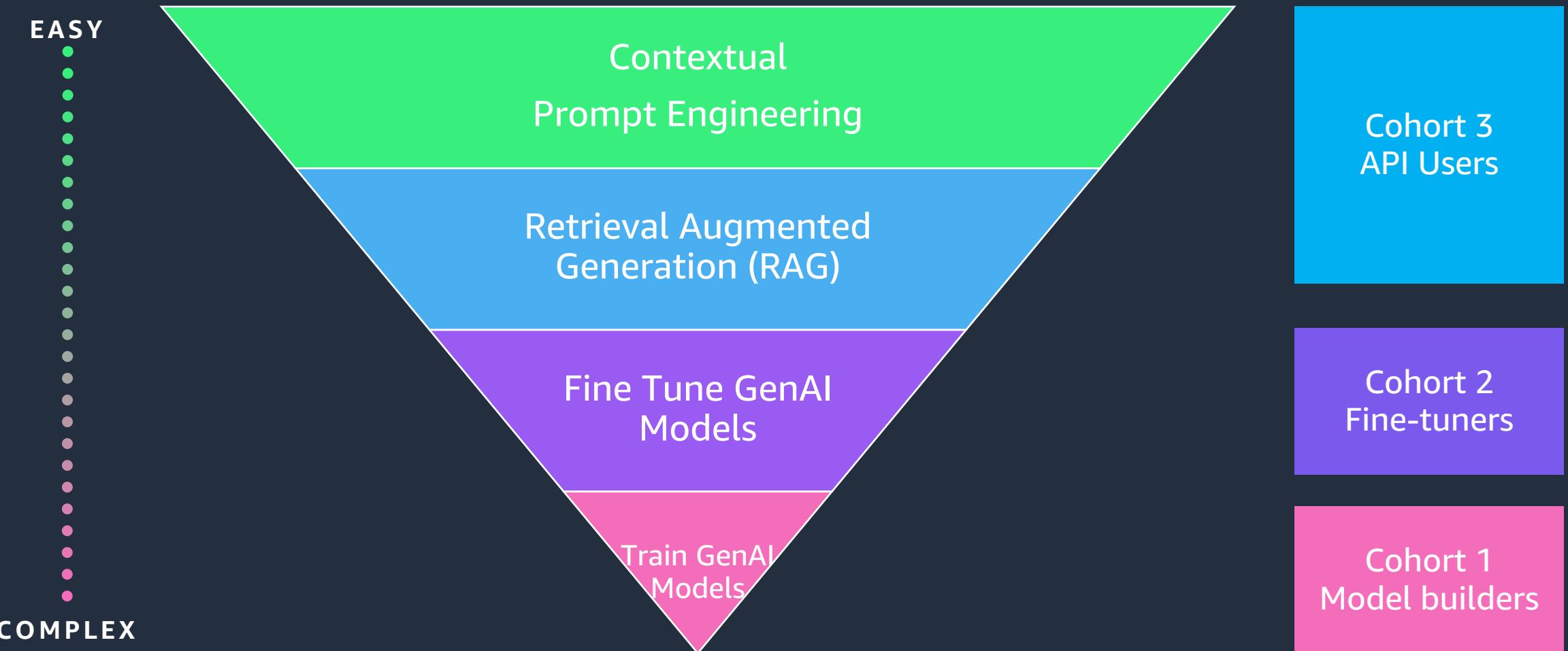
Fine Tune GenAI Models

Train GenAI
Models

LLM customisation patterns & cost



LLM customisation patterns & skills required



상황 별 적절한 Generative AI 활용

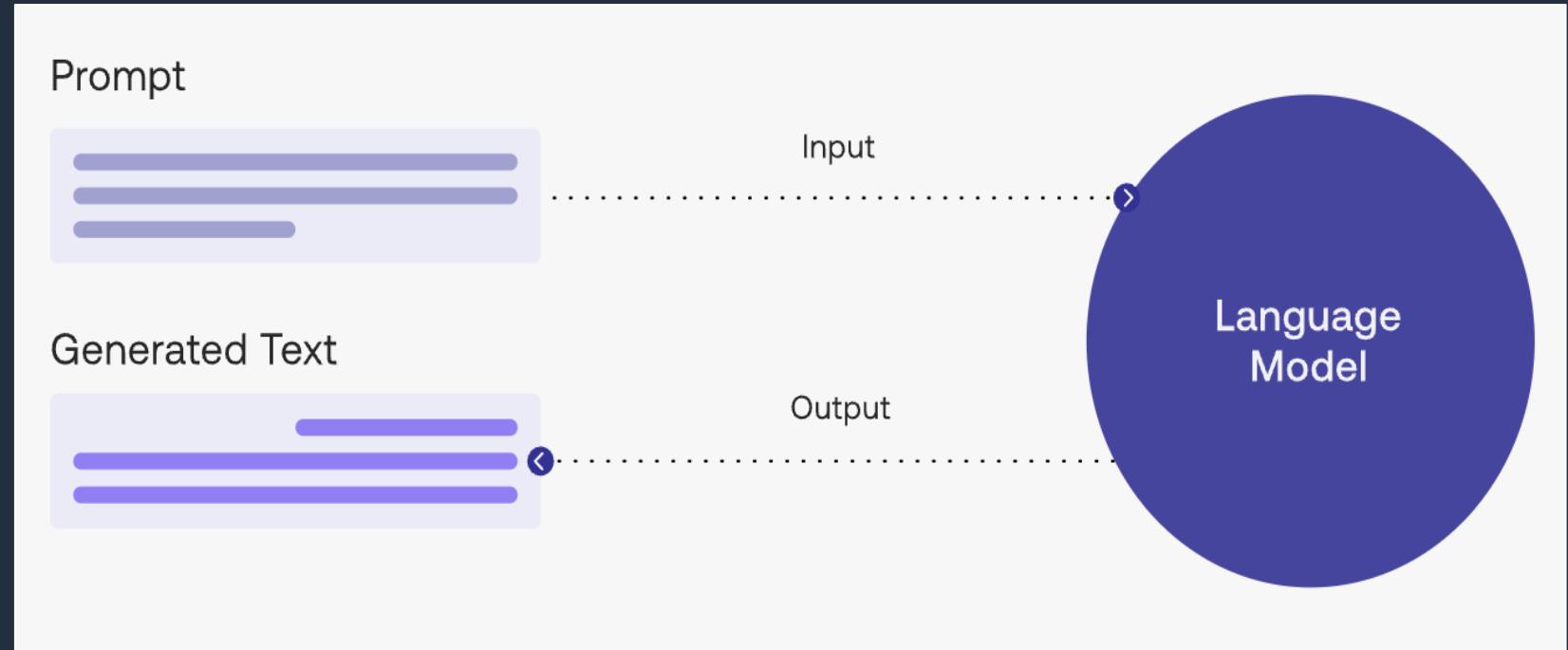
	프롬프트 엔지니어링	검색증강생성(RAG)	파인 투닝	사전 훈련/재훈련
목적	LLM의 유용한 응답을 생성하기 위한 지침/질문/맥락 생성	환각 없는 정확한 응답 생성	응답 품질 및 도메인 관련 결과 개선	모델 공급 (public / proprietary)
훈련 기간	N/A	N/A	대개 몇 분에서 몇 시간	모델 및 인프라에 따라 며칠에서 몇 달까지 소요
비용	CHEAP 			COSTLY
커스터마이징	프롬프트 커스터마이징 (One shot, Few Shot)	기업 내부 데이터를 이용하여 프롬프트 “맥락”을 위한 지식 DB 구축 및 검색	모델 일부 <ul style="list-style-type: none"> 특정 작업 투닝 (Instruction tuning) 도메인별 훈련 데이터 추가 (Domain Adaptation) 	모델 전체 <ul style="list-style-type: none"> 신경망 아키텍처 및 크기 어휘 크기 및 컨텍스트 길이
요구되는 ML 전문성	EASY 			COMPLEX

Prompt Engineering

What is Prompting Engineering?

Prompt Engineering : 생성형 AI모델이 해석하고 이해할 수 있도록 텍스트를 구조화하는 과정

Prompt : AI가 수행해야 할 일을 설명하는 자연어 텍스트



- prompt를 조금만 변경해도 출력에 큰 영향을 미칠 수 있으며, 올바른 출력에 도달하려면 시행착오가 필요합니다.
- prompt는 모델에 대한 *interface* 이므로 프롬프트를 효과적으로 만드는 방법을 아는 것이 중요합니다.

Elements of a Prompt

Instruction

모델 수행 방법에 대한
taks 설명 또는 지침

Summarize the following restaurant review

Context

모델 성능을 조정하기
위한 추가 또는 외부 정보

Restaurant: Luigi's, Location: Naples, Italy, Specialty: Pasta

Input Data

모델이 출력을 제공해야
하는 입력/질문

Review: We were passing through SF on a Thursday afternoon and wanted some Italian food. We passed by a couple places which were packed until finally stopping at Luigi's, mainly because it was a little less crowded and the people seemed to be mostly locals. We ordered the tagliatelle and mozzarella caprese. The tagliatelle were a work of art - the pasta was just right and the tomato sauce with fresh basil was perfect. The caprese was OK but nothing out of the ordinary. Service was slow at first but overall it was fine. Other than that - Luigi's great experience!

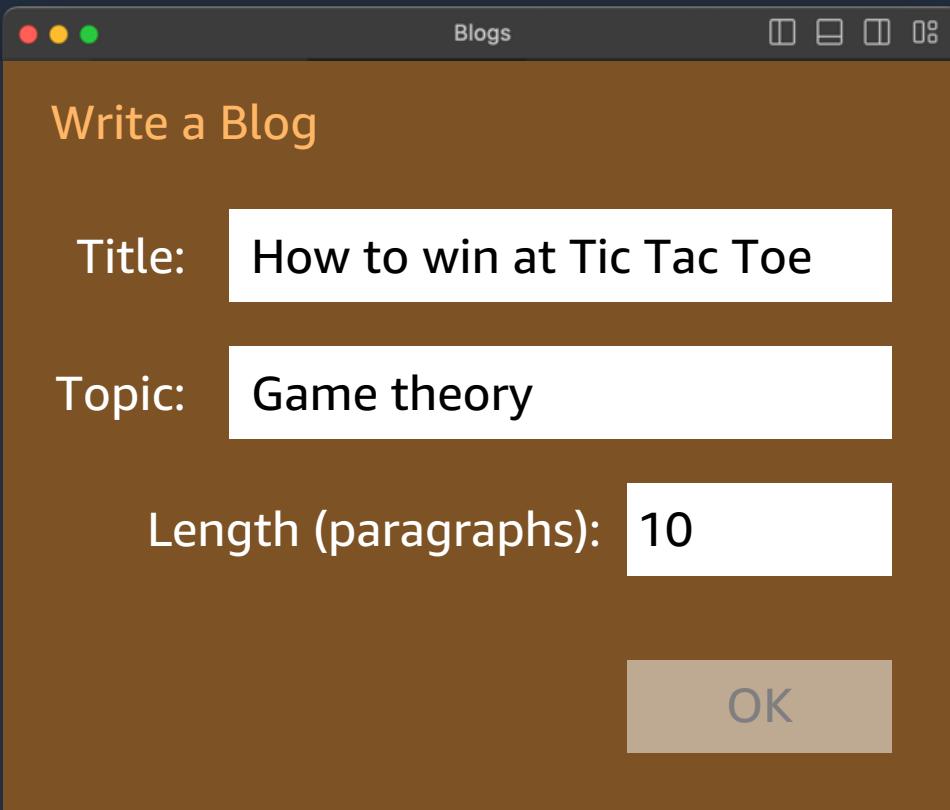
Output Indicator

모델에 출력을 제공하는
데 필요한 지표/형식

Summary: Luigi's is a great Italian restaurant with delicious pasta and good service.



Prompt engineering



"You are a journalist for amazing-acme-blogs.com, an online publication for sophisticated game players aged 18 and up. Articles are written to inform and entertain and represent a unique and non-intuitive perspective.

Write a blog post titled "How to win at Tic Tac Toe" on the topic of game theory. The article should be no longer than 10 paragraphs long.

At the end list 3 prompts that can be used to generate images for the article."

Zero Shot

1. Text Summarization

Input prompt: summarize the text below: text = "Amazon Comprehend uses natural language processing (NLP) to extract insights about the content of documents. It develops insights by recognizing the entities, key phrases, language, sentiments, and other common elements in a document. Use Amazon Comprehend to create new products based on understanding the structure of documents. For example, using Amazon Comprehend you can search social networking feeds for mentions of products or scan an entire document repository for key phrases."

2. Question-Answering prompts

[user] Who are some of the best writers of English literature?

[bot] William Shakespeare, Charles Dickens, Oscar Wilde, Jane Austen, and William Blake

[user] Can you tell me more about Wilde?

[bot] He was a poet, painter, and engraver.

[user] Where was he born, and when?

[bot] Blake was born in London in 1757.

[user] What is the most famous book by him? what is the rationale behind it?

[bot]



One Shot & Few Shot Prompting

1. Indicate topics

Indicate the topic from the Message below:

Message: When the spaceship landed on Mars, the whole humanity was excited

Topic: space

Message: I am trying to cook tofu with tomatoes.

Topic:

2. Sentimental Classification

Classification:

Tweet: "I hate it when my phone battery dies.": Sentiment: Negative

Tweet: "My day has been great": Sentiment: Positive

Tweet: "This is the link to the article": Sentiment: Neutral

Tweet: "This new music video was incredibile" Sentiment:

Chain of Thought Prompting

Chain-of-thought prompting

Standard Prompting

Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

A: The answer is 27.

Chain-of-Thought Prompting

Model Input

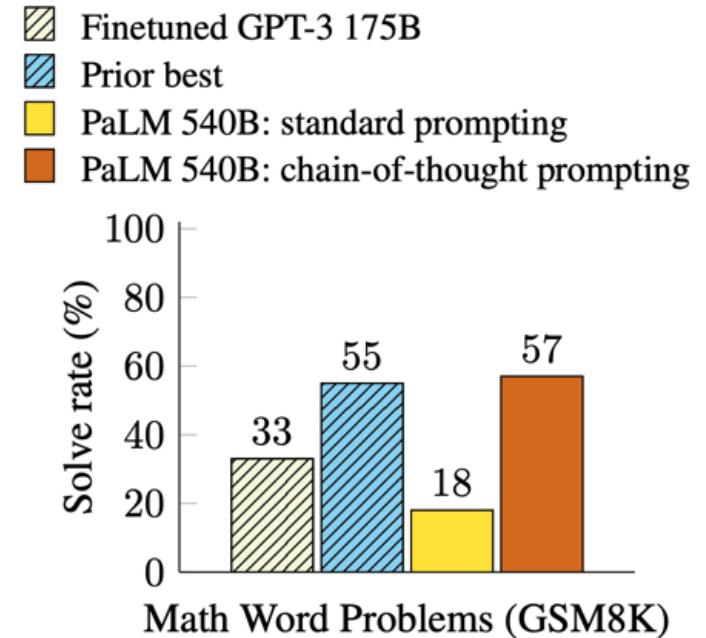
Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. $5 + 6 = 11$. The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

A: The cafeteria had 23 apples originally. They used 20 to make lunch. So they had $23 - 20 = 3$. They bought 6 more apples, so they have $3 + 6 = 9$. The answer is 9.



Prompting 평가

Input

Output

In-Context Learning

Q : 리사는 비누를 5개 가지고 있습니다. 그녀는 6개 비누가 들어 있는 세트를 2개 더 구입합니다. 비누를 얼마나 많이 가지고 있나요?

A : 정답은 17입니다.

Q : 카페테리아는 37개의 바나나가 있습니다. 그들은 바나나가 5개씩 들어 있는 5개 묶음을 더 샀는데, 얼마나 많은 바나나가 있나요?

A : 정답은 62입니다.

Instruction-Following

여기 수학적 추론 문제가 있습니다. 정답을 만들기 위해서는 산술 연산을 적용해야 합니다.

Q : 리사는 비누를 5개 가지고 있습니다. 그녀는 6개 비누가 들어 있는 세트를 2개 더 구입합니다. 비누를 얼마나 많이 가지고 있나요?

A : 정답은 17입니다.

Q : 카페테리아에는 37개의 바나나가 있습니다. 그들은 5개씩 들어 있는 ~~

A : 정답은 62입니다.

Chain-of-Thought

Q : 리사는 비누를 5개 가지고 있습니다. 그녀는 6개 비누가 들어 있는 세트를 2개 더 구입합니다. 비누를 얼마나 많이 가지고 있나요?

A : 리사는 5개 비누로 시작합니다. 비누가 6개씩 들어있는 2개의 세트는 12개의 비누가 있습니다. $5+12=17$. 정답은 17입니다.

Q : 카페테리아에는 37개의 바나나가 있습니다. 그들은 바나나가 5개씩 ~~

A : 카페테리아는 원래 37개 바나나가 있습니다. 5개 묶음을 더 구매했고, 각 묶음에는 5개씩 들어있으므로, 재고는 $5 \times 5 = 25$ 바나나를 추가했습니다. 우리는 이 숫자를 더합니다. $37 + 25 = 62$. 정답은 62입니다.



prompt 효과

Context

input

Output

조선왕조실록에 기록 된 세종대왕 맥북
프로 던짐 사건에 대해 알려줘

세종대왕이 맥북 프로를 던지는 사건은
조선왕조실록에 기록되어 있습니다. 이
사건은 1443년(세종 25년) 세종이
집현전 학자들과 함께 궁중 후원에서
열린 연회에서 벌어진 일입니다. 세종은
연회 도중에 갑자기 맥북 프로를 바닥에
던져버렸습니다. 이 사건은 세종의
권위에 도전하는 것으로 받아들여졌고,
세종은 연회를 중단하고 신하들에게
맥북 프로를 가져오라고 명령했습니다.

~~~~~

'조선왕조실록'에는 세종대왕 맥북프로  
던짐 사건이 기록된 바 없습니다.

조선왕조실록에 기록 된 세종대왕 맥북  
프로 던짐 사건에 대해 알려줘

'조선왕조실록'에는 세종대왕이  
맥북프로를 던지는 사건이 기록되어  
있지 않습니다. 그러나 이 사건은  
세종대왕이 맥북프로를 던지는 것을  
금지하는 명령을 내린 것으로 알려져  
있습니다. 이 명령은 세종대왕이  
맥북프로를 던지는 것이 국가의 이익에  
반하는 것으로 간주되었기 때문에  
내려졌습니다. ~~~~

현대의 사물인 '맥북프로'가 등장할 수  
없습니다.

조선왕조실록에 기록 된 세종대왕 맥북  
프로 던짐 사건에 대해 알려줘

조선왕조실록에 기록된 세종대왕이  
맥북프로를 던지는 사건은 맥북프로가  
현대의 사물이 아니기 때문에 발생할 수  
없습니다. 맥북프로는 애플에서 만든  
노트북 컴퓨터로, 현대의 사물이  
아닙니다.



# Hallucination 억제

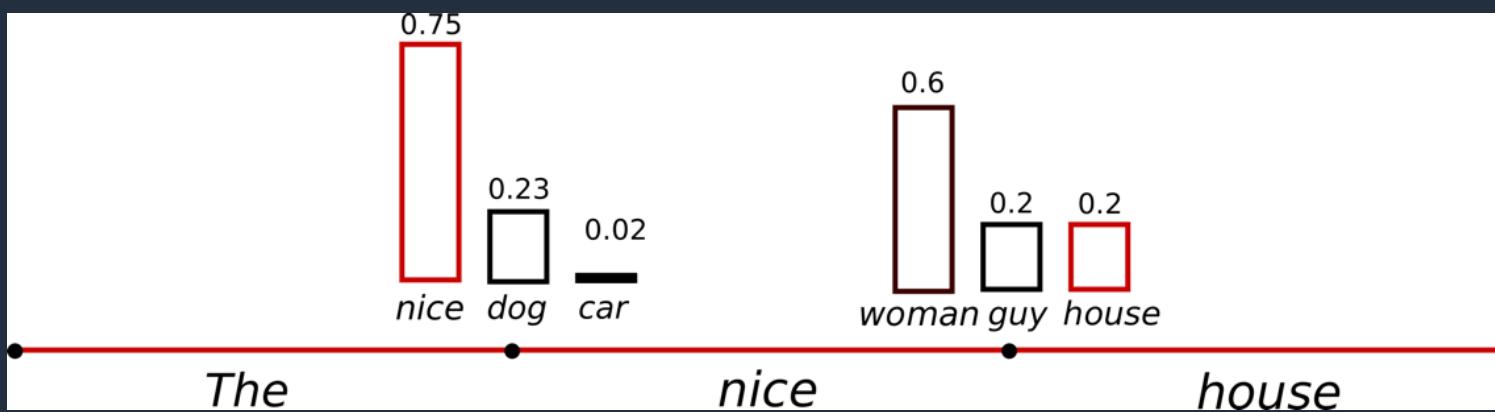
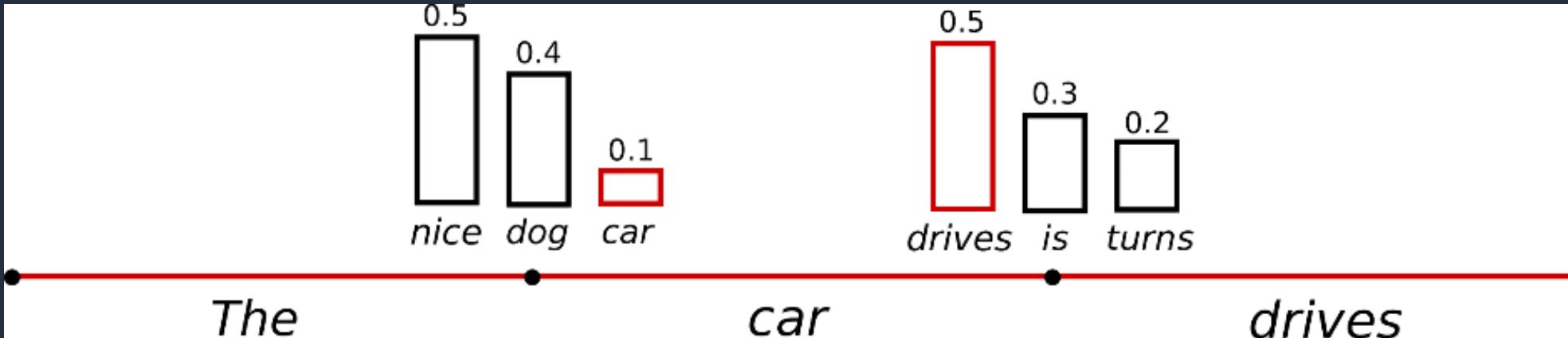
LLM say "I don't know" to prevent hallucinations

Human: Answer the following question only if you know the answer or can make a well-informed guess; otherwise tell me you don't know it.

What was the heaviest hippo ever recorded?

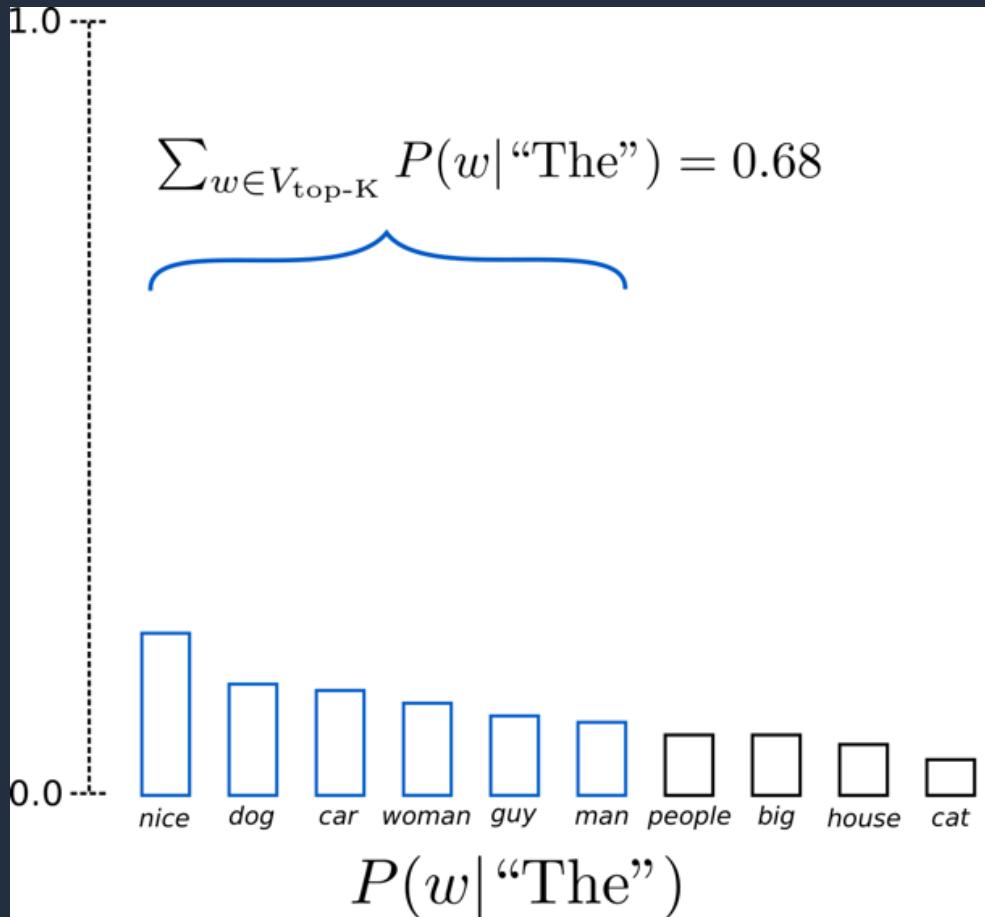
Assistant:

# Temperature

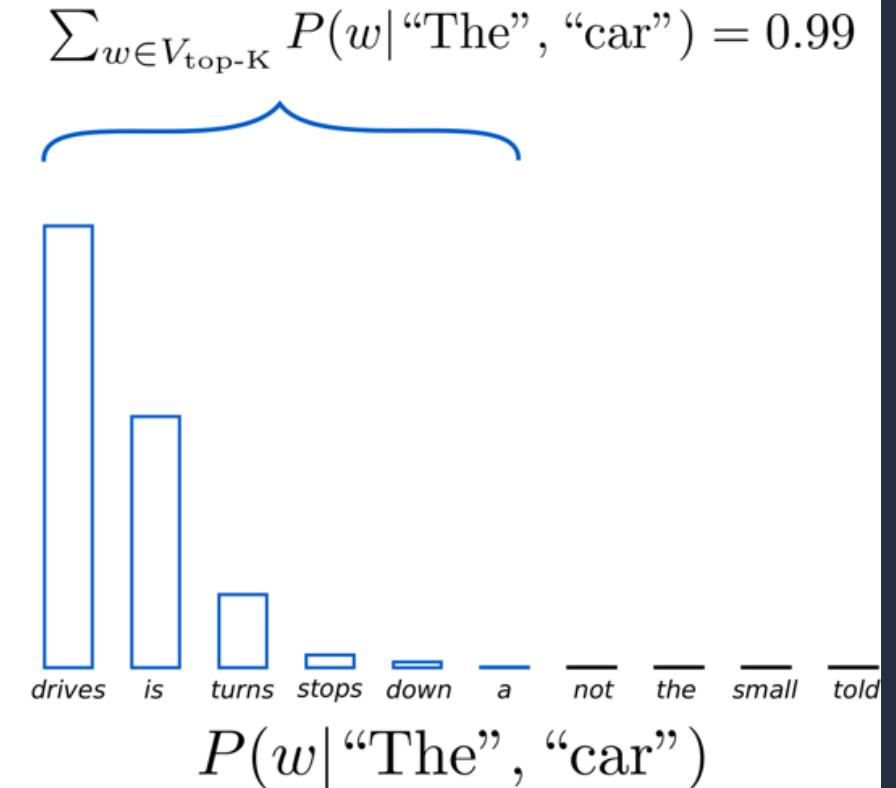


<https://huggingface.co/blog/how-to-generate>

# Top\_k

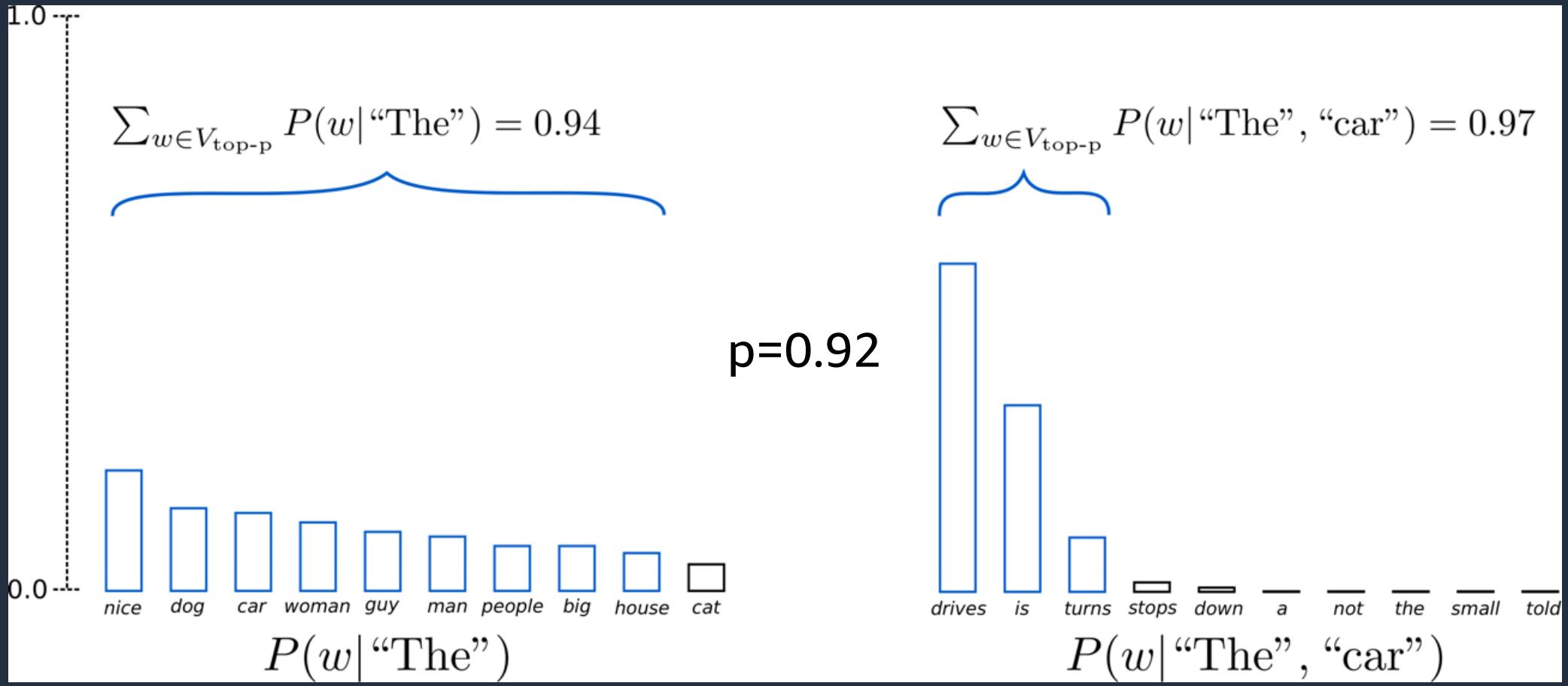


k=6



<https://huggingface.co/blog/how-to-generate>

# Top\_p



<https://huggingface.co/blog/how-to-generate>

# RAG (Retrieval-Augmented Generation)

# RAG 설명 - 특정 지식을 위한 LLM



XX 회사 6월 매출은?

XX 회사 6월 매출은  
약 12억원입니다.  
“한각 발생”

LLM은 특정 지식이 아닌 일반적인 추론을 위한 것입니다.

# RAG 설명 - 특정 지식을 위한 LLM



**Context**는 LLM에 특정 지식 및 최신 정보를 제공하는 방법입니다.

# [Again] What are large language models (LLMs)?

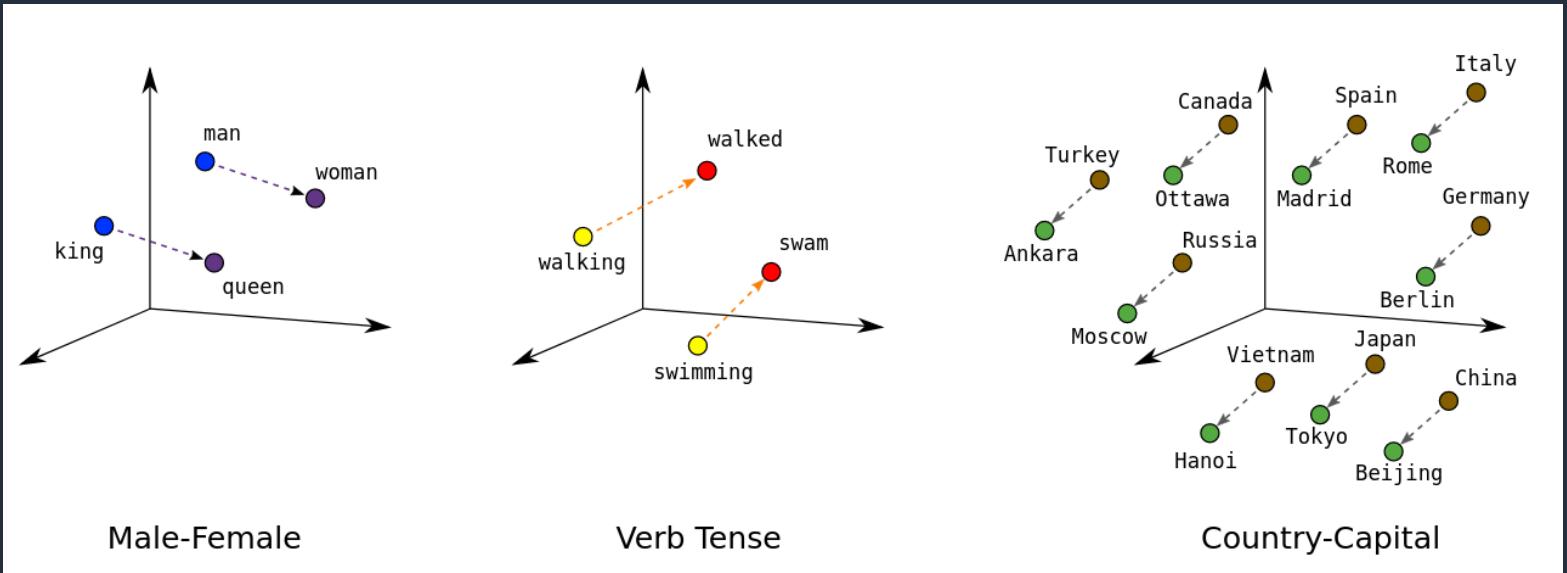


# Embedding

사람이 쓰는 자연어를 기계가 이해할 수 있는 숫자의 나열인 벡터로 바꾼 결과 혹은 그 과정 혹은 전체를 의미  
텍스트를 실수 벡터 형태(I.E. FLOATING POINT 숫자들로 구성된 고정된 크기의 배열)로 표현한 결과물

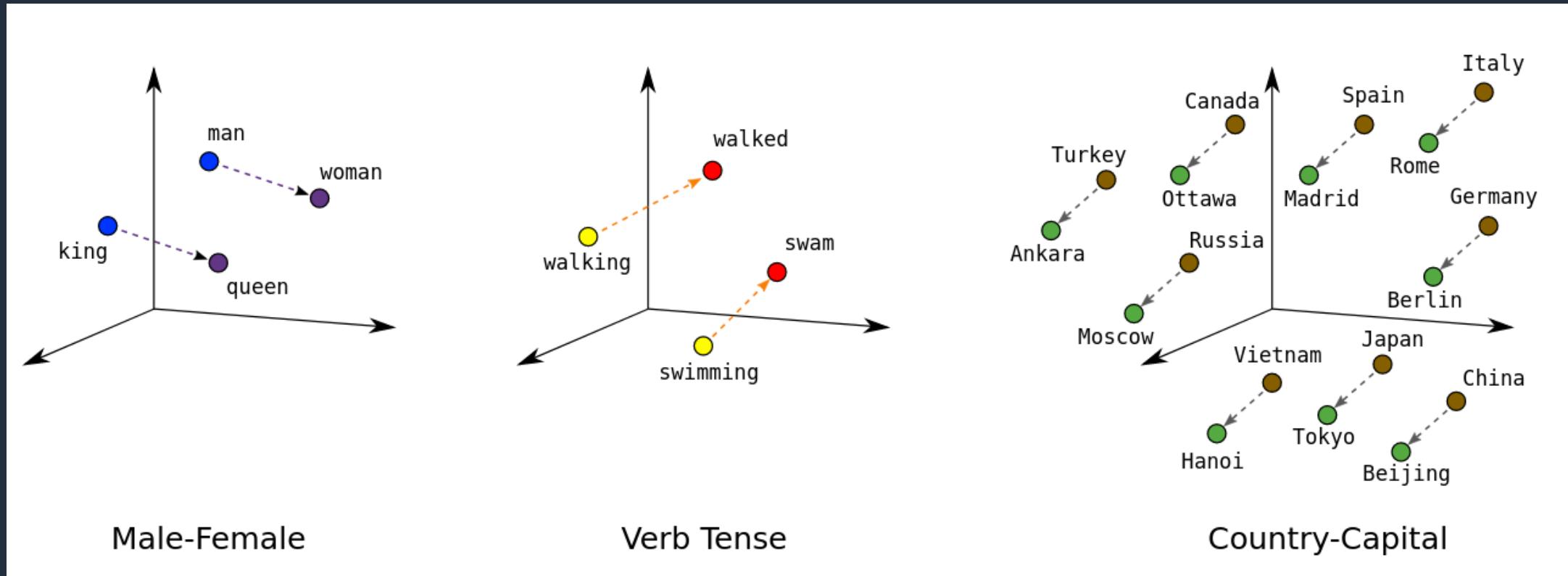
Cheese =

$$\begin{pmatrix} 0.275 \\ 0.827 \\ -0.133 \\ 0.298 \\ 0.023 \\ -0.394 \\ 0.271 \\ 0.112 \end{pmatrix}$$



벡터간 유사도를 통해 단어들의 의미 확인

# Latent space / Latent Feature Space / Embedding Space



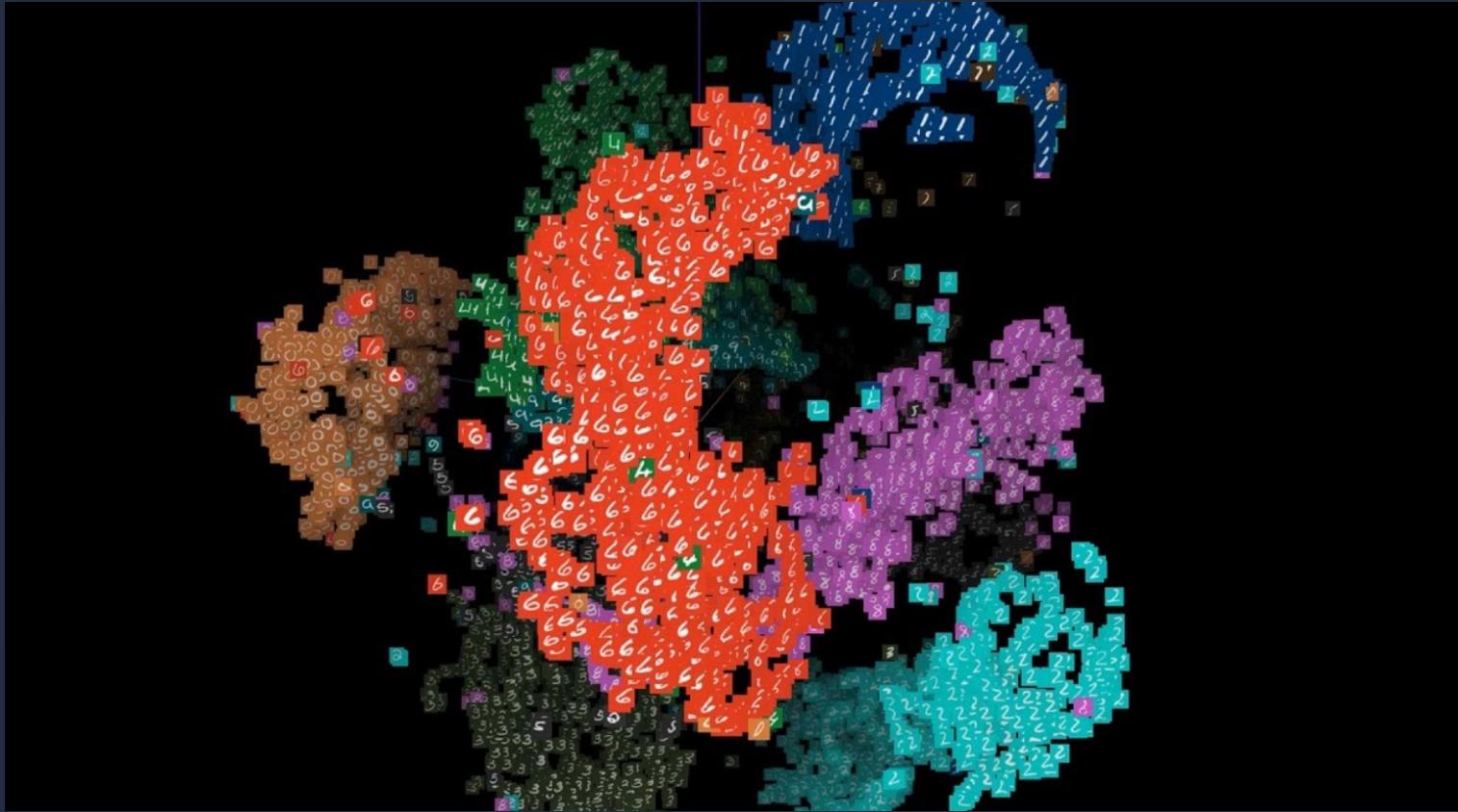
$$\text{cosine similarity} = S_C(A, B) := \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}},$$

```
function cosineSimilarity(a: Array<Number>, b: Array<Number>) {
    return dotProduct(a, b) / (length(a) * length(b))
}
function dotProduct(a: Array<Number>, b: Array<Number>) {
    return zip(a, b).map((pair) => pair[0] * pair[1]).sum()
}
function length(a: Array<Number>) {
    return sqrt(a.sum(s => Math.pow(s, 2)));
}
```



# High Dimensional Space

Latent space / Latent Feature Space / Embedding Space



A.I. Experiments: Visualizing High-Dimensional Space

<https://experiments.withgoogle.com/visualizing-high-dimensional-space>

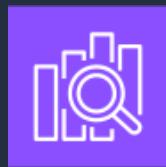
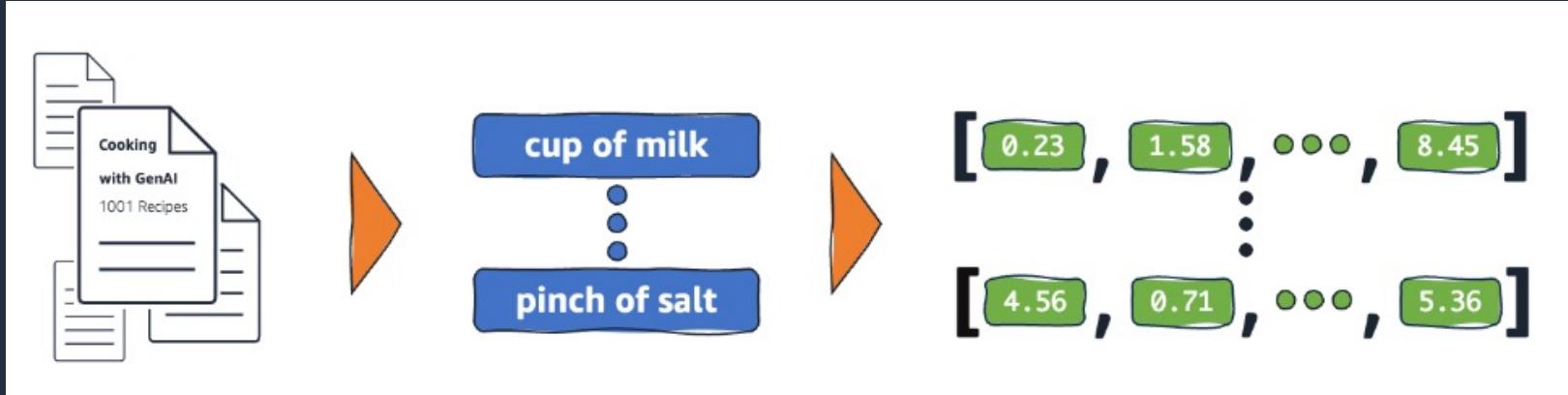


© 2023, Amazon Web Services, Inc. or its affiliates.

© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# Vector Datastores

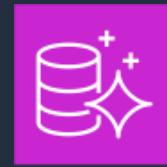
최대 수십 억 개의 임베딩(벡터)를 효율적으로 저장, 비교 및 검색하는 역할



Amazon OpenSearch  
With K-NN



Amazon RDS  
With pgvector



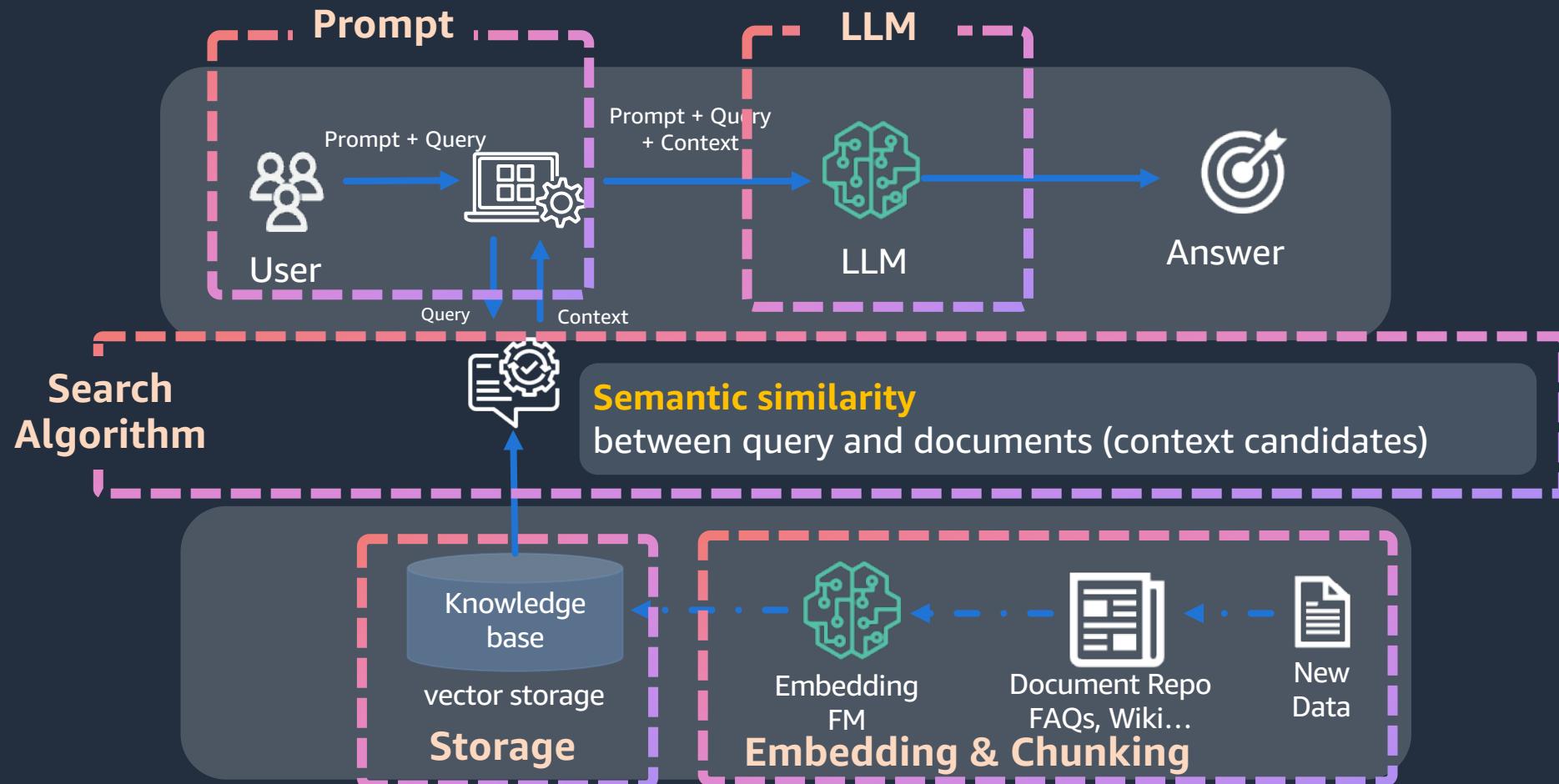
Amazon Aurora  
With pgvector



Amazon Kendra



# RAG의 구성요소들



# RAG vs. PEFT

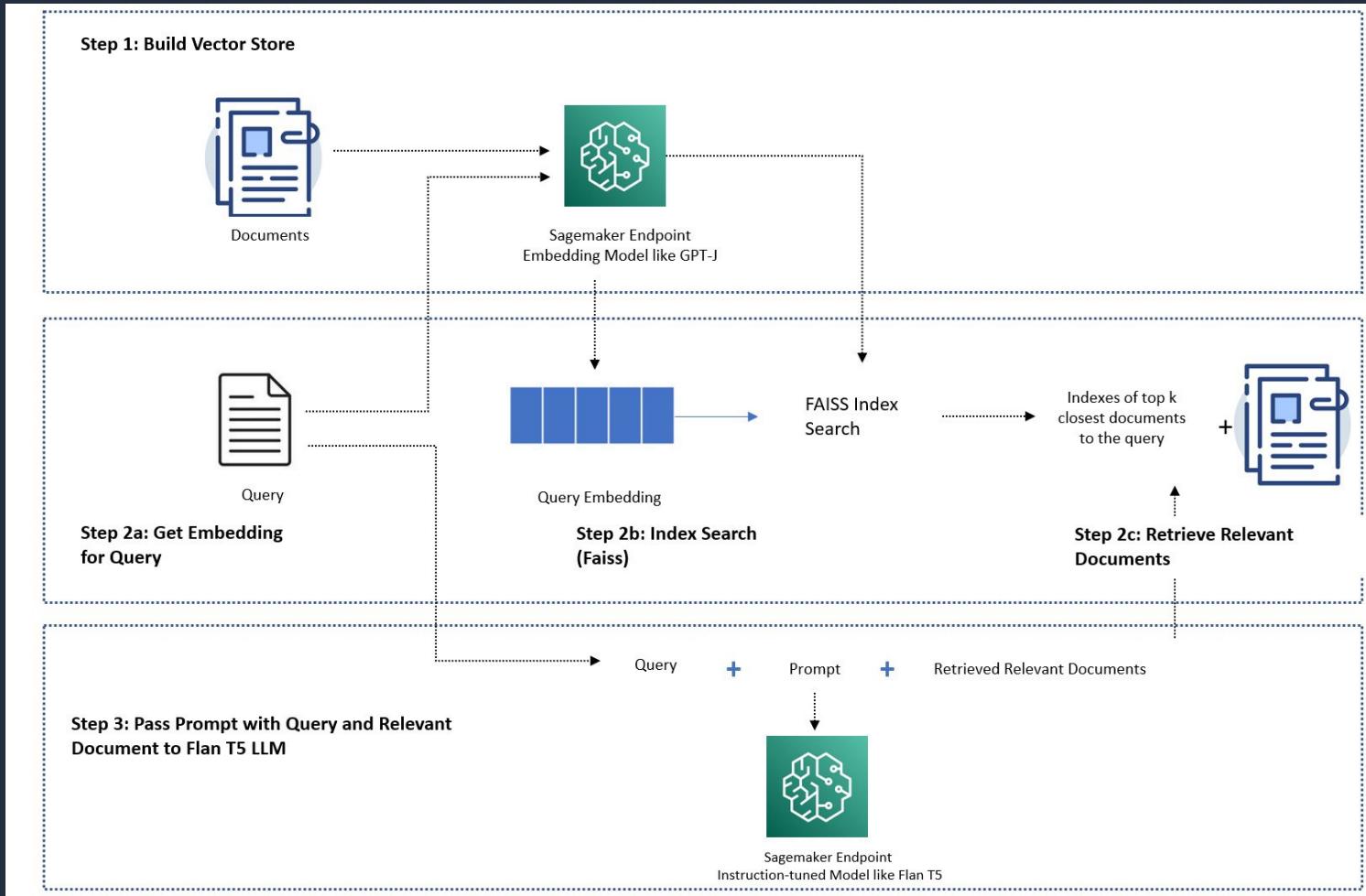
## RAG

- 빠른 구현
- 최신 데이터 적용 용이
- 사용자 지정 데이터가 네트워크 내에 유지됨 (프롬프트의 컨텍스트 제외)
- Semantic Search의 고가용성 필요
- 모든 종류의 작업에 유효하지 않음
- 컨텍스트 길이 제한으로 문서에 대한 심층적인 내용 분석 어려움

## PEFT

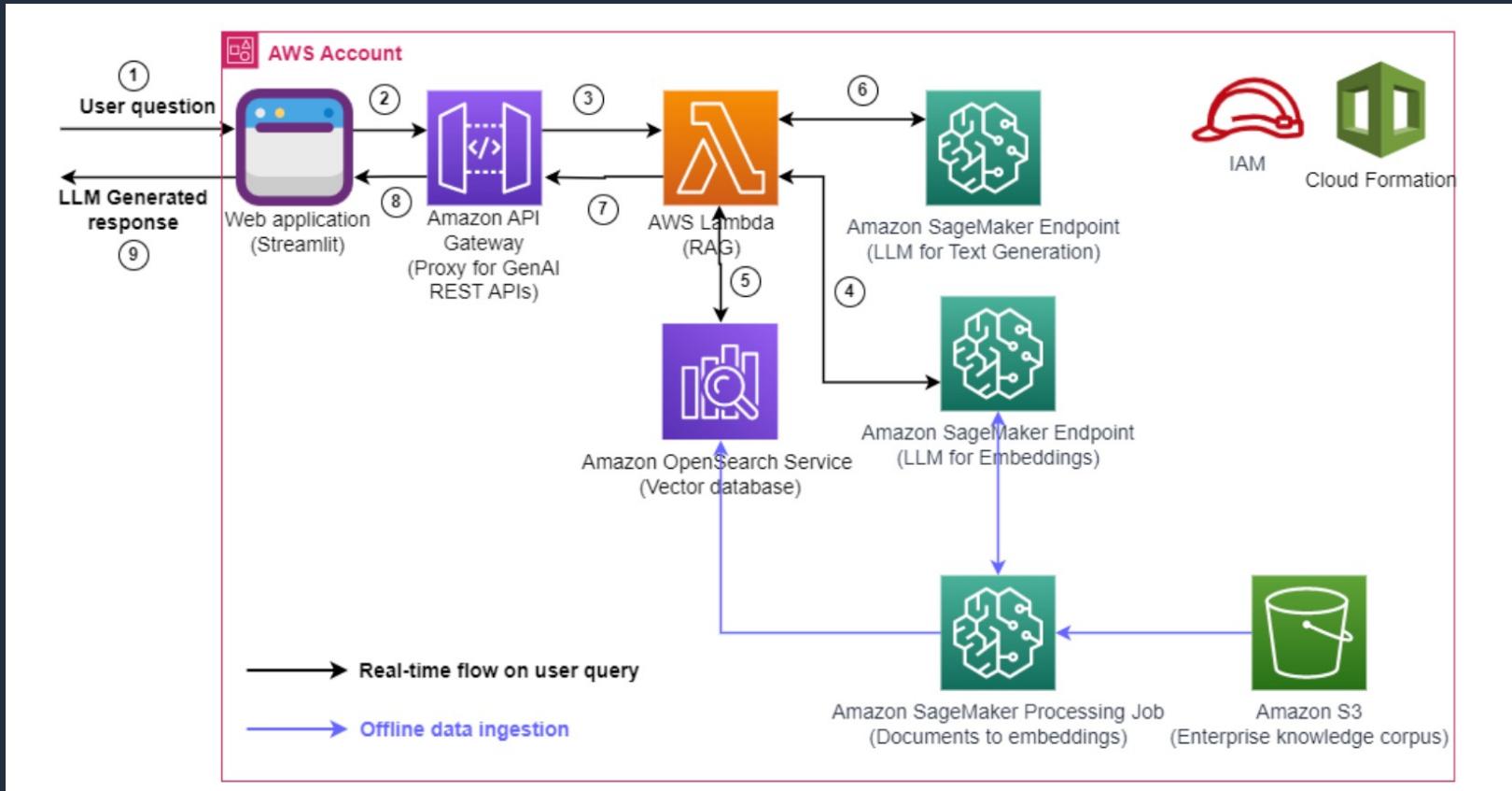
- ML 인력, 컴퓨팅 리소스 필요
- 최신 데이터 유지의 어려움
- 클라우드 리소스 사용 시 네트워크 외부 경유 필요
- 학습에 GPU 자원 필요
- 작업 유형에 적합한 데이터 필요
- 프롬프트 엔지니어링 경감
- 도메인 특화 작업에 유효
- 더 심층적인 질문에 답할 수 있음
- 작은 LLM으로 좋은 결과

# RAG using Sagemaker KNN



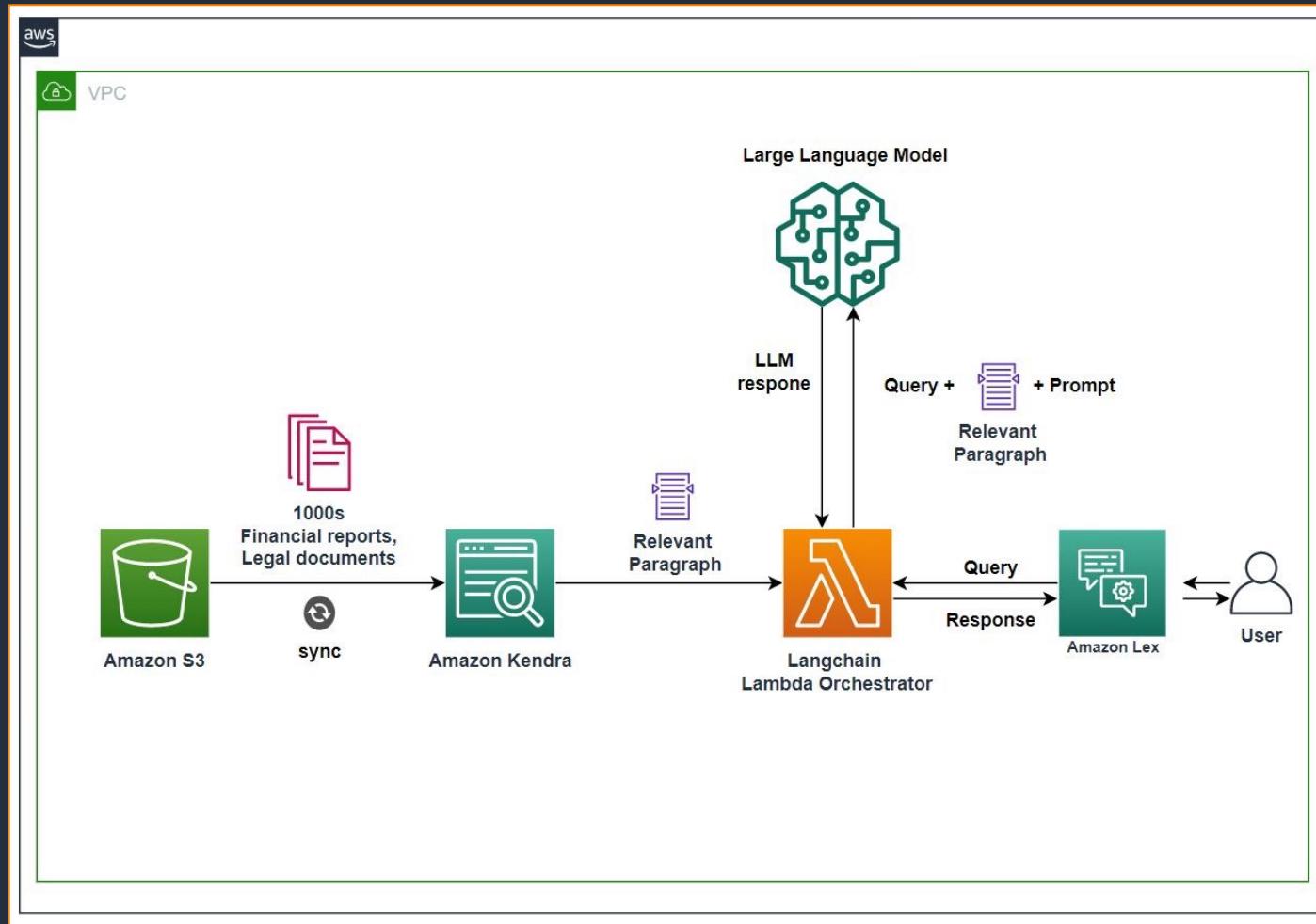
[https://sagemaker-examples.readthedocs.io/en/latest/introduction\\_to\\_amazon\\_algorithms/jumpstart-foundation-models/question\\_answering\\_retrieval\\_augmented\\_generation/question\\_answering\\_jumpstart\\_knn.html](https://sagemaker-examples.readthedocs.io/en/latest/introduction_to_amazon_algorithms/jumpstart-foundation-models/question_answering_retrieval_augmented_generation/question_answering_jumpstart_knn.html)

# RAG using API Gateway, Lambda and OpenSearch



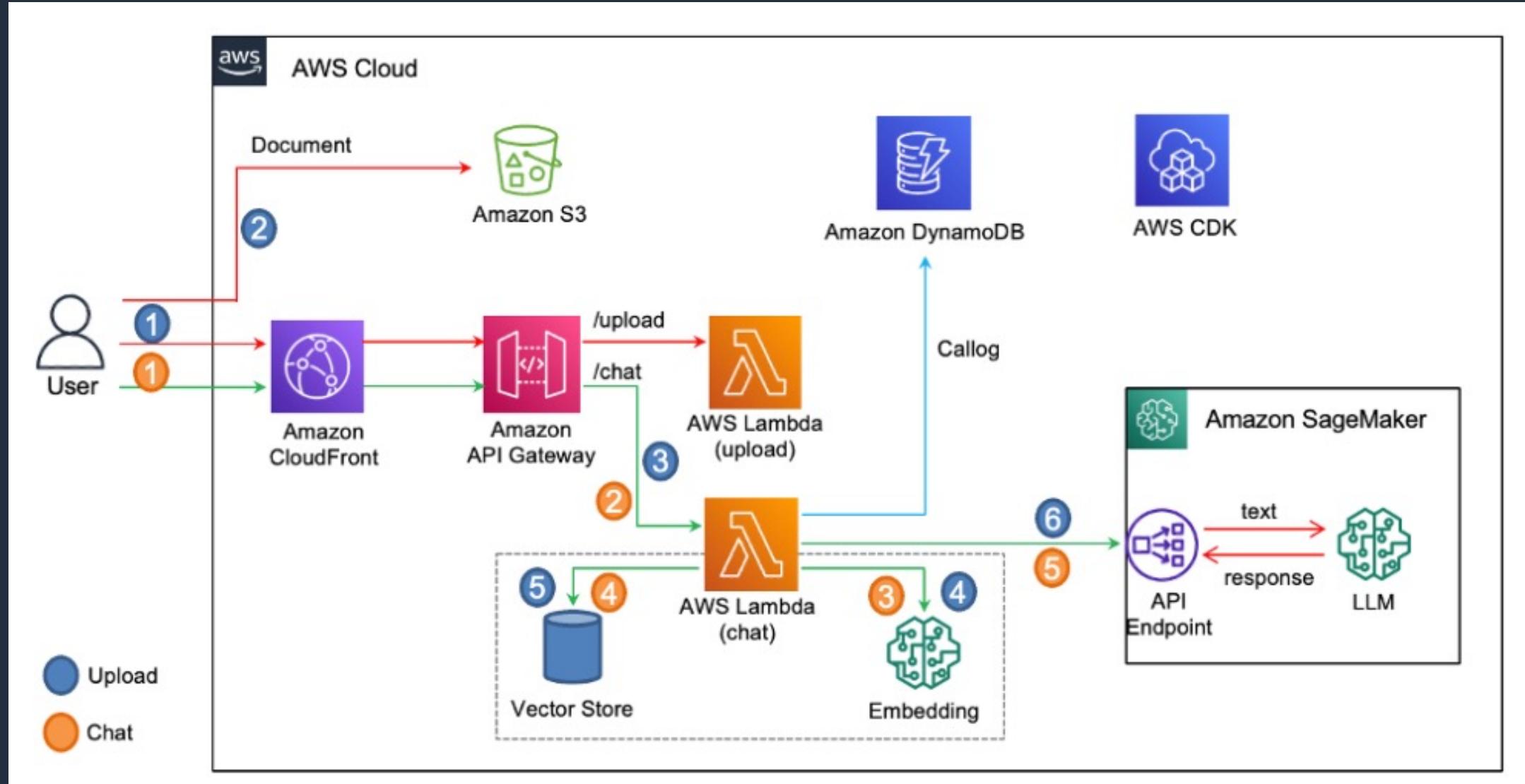
<https://aws.amazon.com/ko/blogs/machine-learning/build-a-powerful-question-answering-bot-with-amazon-sagemaker-amazon-opensearch-service-streamlit-and-langchain/>

# RAG with Kendra and Langchain



<https://aws.amazon.com/ko/blogs/tech/quickly-build-high-accuracy-generative-ai-applications-on-enterprise-data-using-amazon-kendra-langchain-and-large-language-models/>

# Vector Store with Llama2 on SageMaker JumpStart



# 참고자료

## RAG on Amazon

<https://aws.amazon.com/blogs/machine-learning/quickly-build-high-accuracy-generative-ai-applications-on-enterprise-data-using-amazon-kendra-langchain-and-large-language-models/>

<https://aws.amazon.com/blogs/machine-learning/build-a-powerful-question-answering-bot-with-amazon-sagemaker-amazon-opensearch-service-streamlit-and-langchain/>

<https://aws.amazon.com/blogs/machine-learning/dialogue-guided-intelligent-document-processing-with-foundation-models-on-amazon-sagemaker-jumpstart/>

<https://aws.amazon.com/blogs/machine-learning/question-answering-using-retrieval-augmented-generation-with-foundation-models-in-amazon-sagemaker-jumpstart/>

## Amazon Kendra

<https://docs.aws.amazon.com/kendra/latest/dg/what-is-kendra.html>

<https://docs.aws.amazon.com/kendra/latest/dg/tutorial-search-metadata.html>

## Repositories

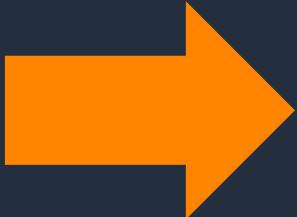
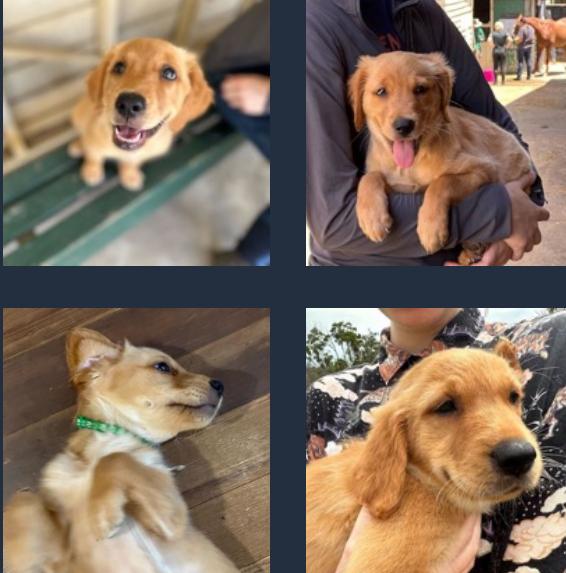
<https://github.com/aws-samples/amazon-kendra-langchain-extensions>



# Fine Tuning

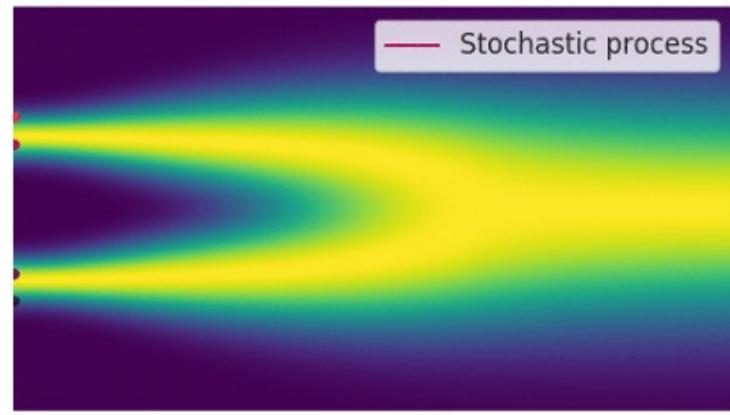
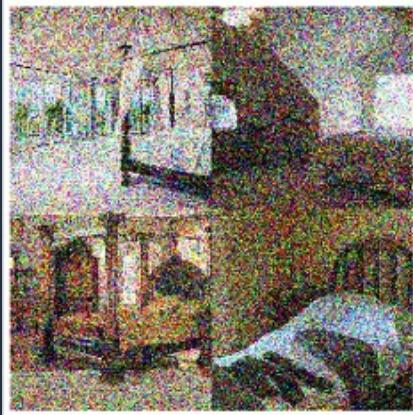
# Stable Diffusion Fine Tuning Examples

Diffusion models.

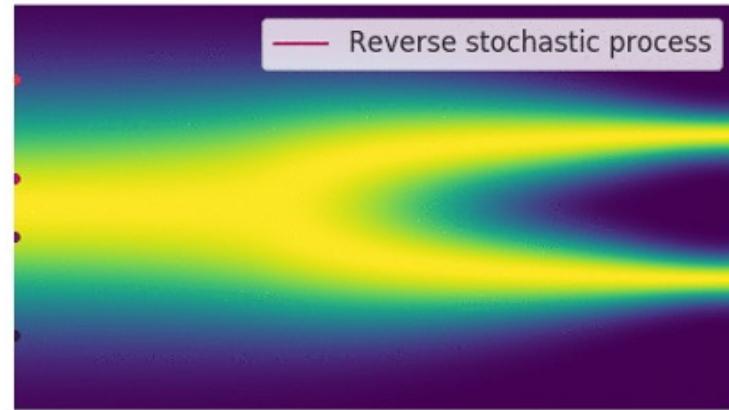
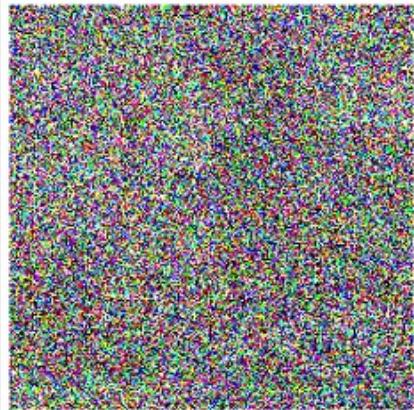


"Marigold the puppy"

# Brief Overview of Diffusion Models



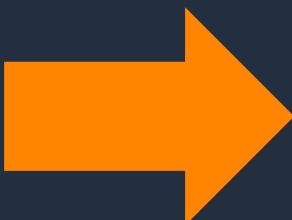
destroy the data by gradually adding small amounts of gaussian noise



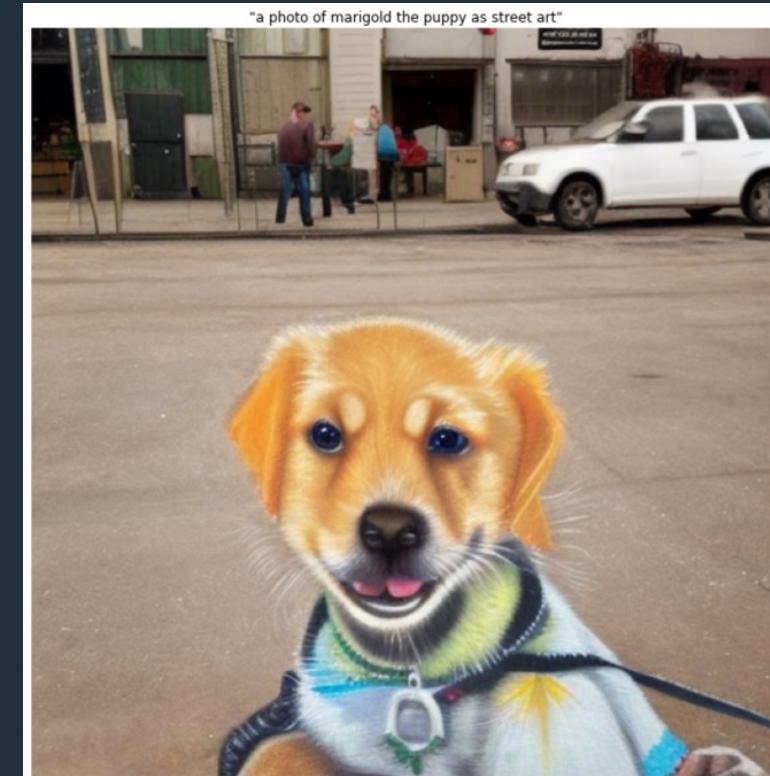
create data by gradually denoising a noisy code from a stationary distribution

# Stable Diffusion Fine Tuning Examples

Diffusion models.



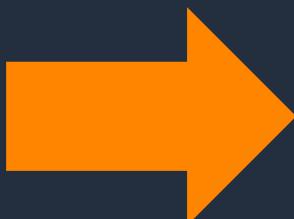
"**d the puppy"**



**"a photo of Marigold the  
puppy as street art"**

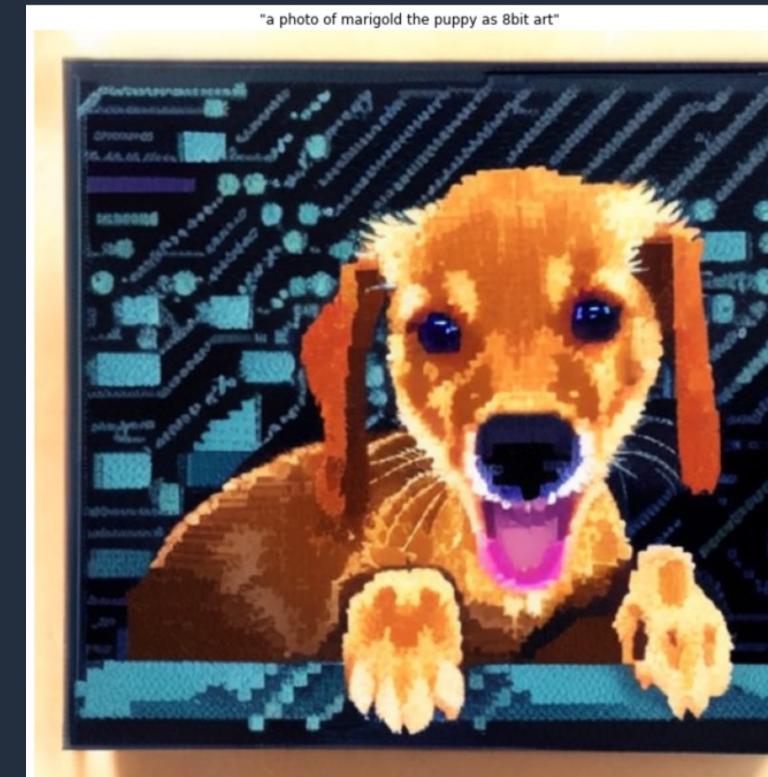
# Stable Diffusion Fine Tuning Examples

Diffusion models.



Stable  
Diffusion  
Fine Tuning

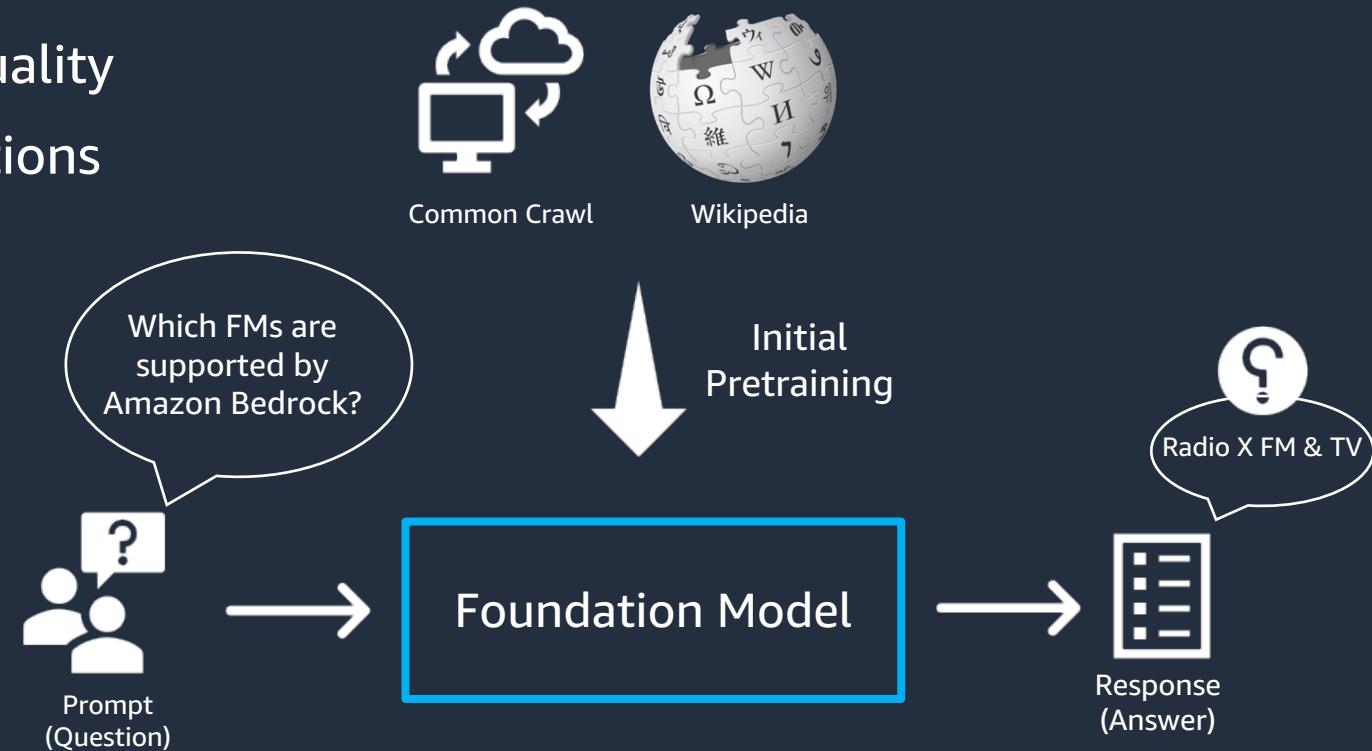
d the puppy"



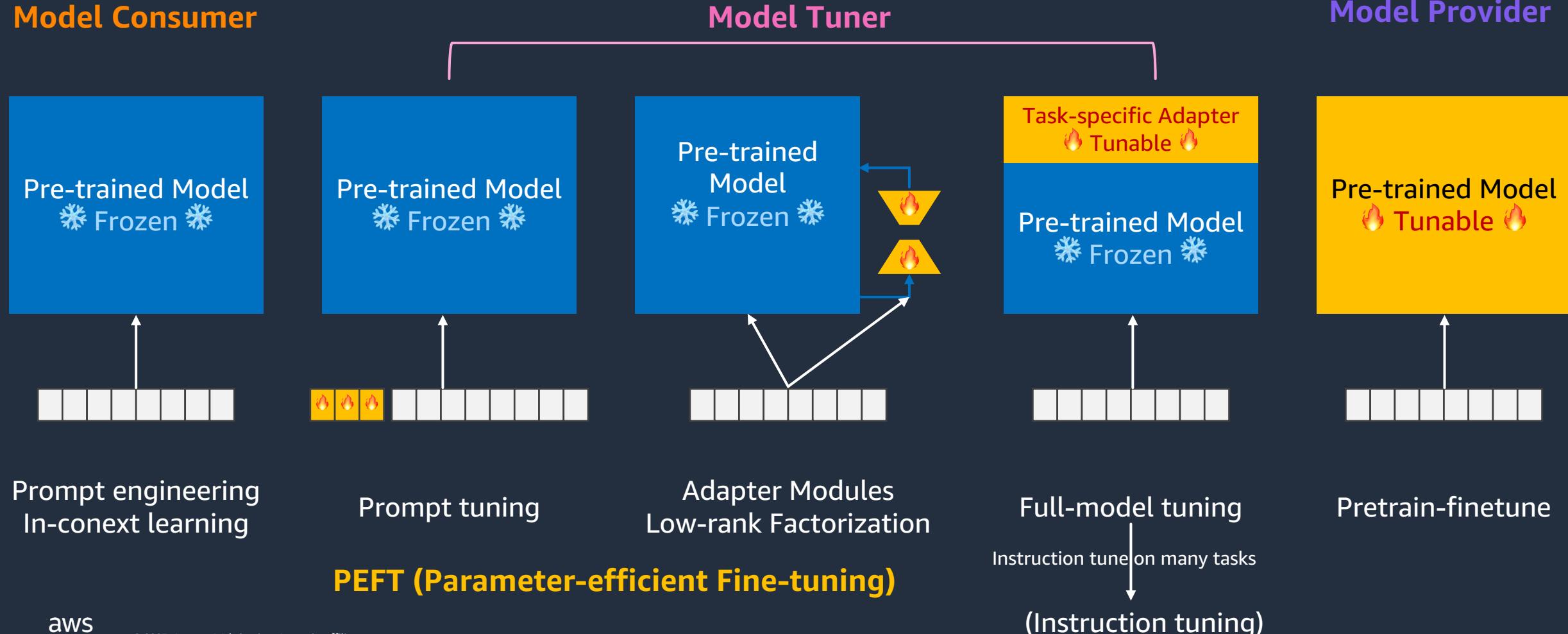
"a photo of Marigold the  
puppy as 8bit art"

# Why customize a foundation model?

- Specific Task
- Closed-domain knowledge
- Current Knowledge
- Improving the performance/quality
- Reduce likelihood of hallucinations
- **Out of Memory**

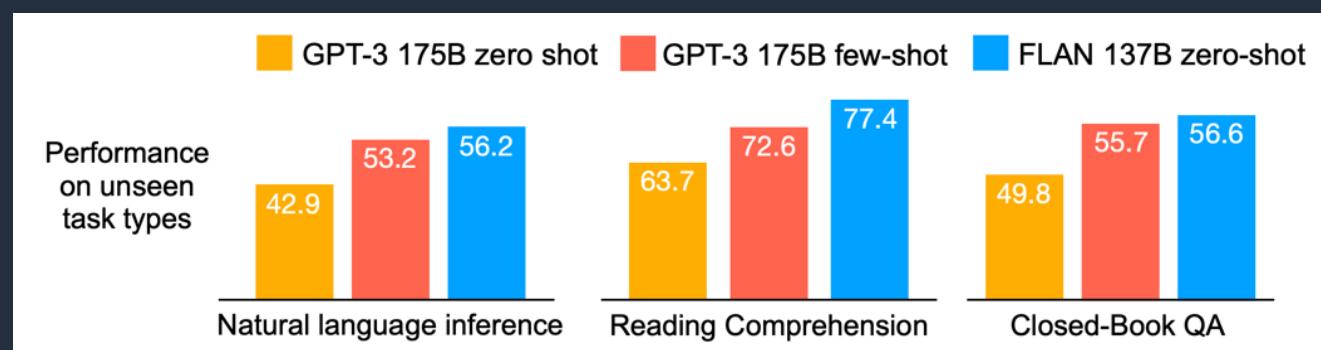
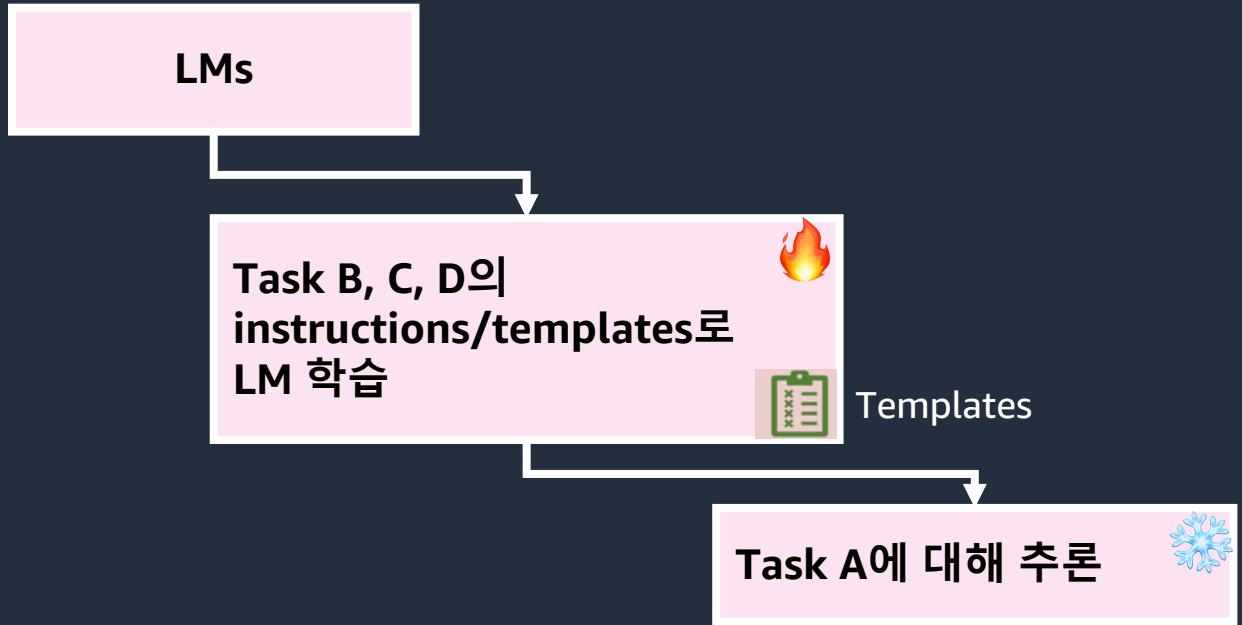


# Prompt Engineering and Fine-tuning



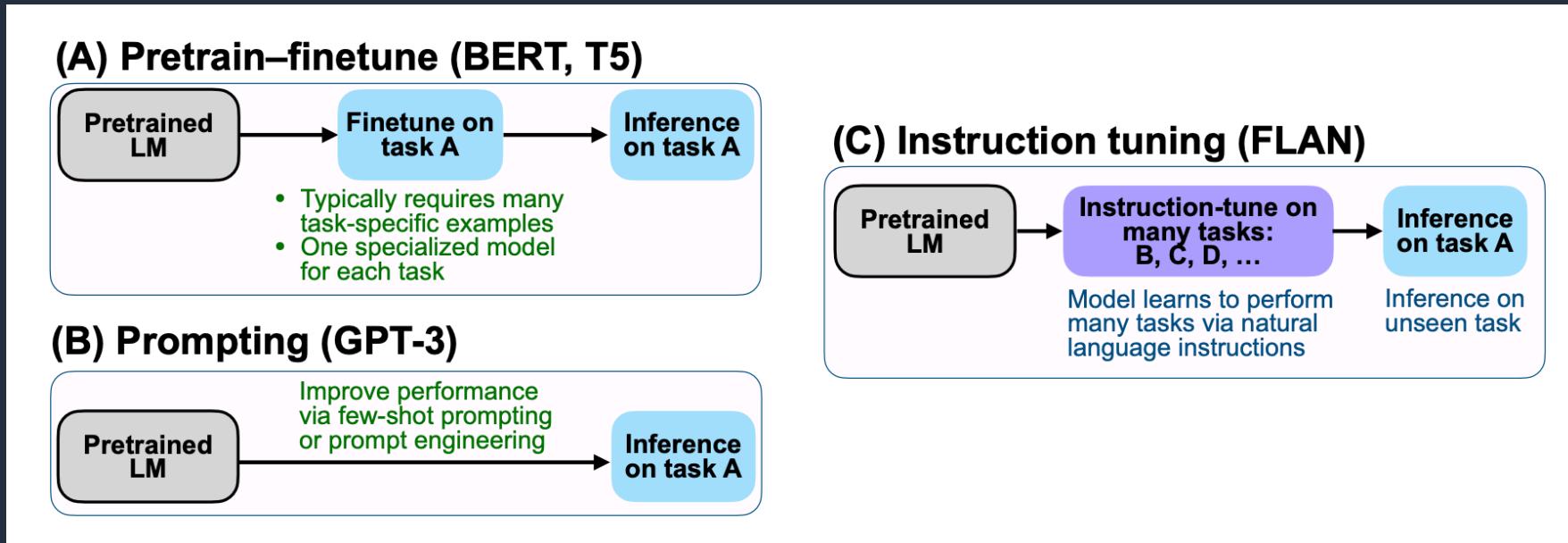
# Instruction tuning

- Instruction을 통해 설명된 데이터 세트 모음에서 언어 모델을 미세 조정
- Unseen task에서 zero shot 성능 개선
- Instruction tuning은 Instruction으로 표현된 mixture data로 사전 학습된 언어 모델을 튜닝하며, 추론 시에는 unseen 작업 유형을 평가
- 3가지 unseen 작업 유형에 대해, zero-shot, few-shot GPT-3과 비교하여 zero-shot FLAN (Finetuned Language Net)의 성능을 평가
- 작은 LLM에서도 좋은 성능



Ref : Wei, Jason, et al. "Finetuned language models are zero-shot learners." arXiv preprint arXiv:2109.01652 (2021).

# Instruction tuning comparison



```
[{'instruction': '건강을 유지하기 위한 세 가지 팁을 알려주세요.',
  'input': '',
  'output': '세 가지 팁은 아침식사를 꼭 챙기며, 충분한 수면을 취하고, 적극적으로 운동을 하는 것입니다.'},
 {'instruction': '세 가지 기본 색은 무엇인가요?',
  'input': '',
  'output': '기본 색은 빨강, 파랑, 노랑입니다.'},
 {'instruction': '원자의 구조를 설명하세요.',
  'input': '',
  'output': '원자는 양성자, 중성자, 전자로 구성되어 있으며, 양성자와 중성자는 원자핵 안에 있고 전자는 주변에 있습니다.'},
```

Ref : Wei, Jason, et al. "Finetuned language models are zero-shot learners." arXiv preprint arXiv:2109.01652 (2021).

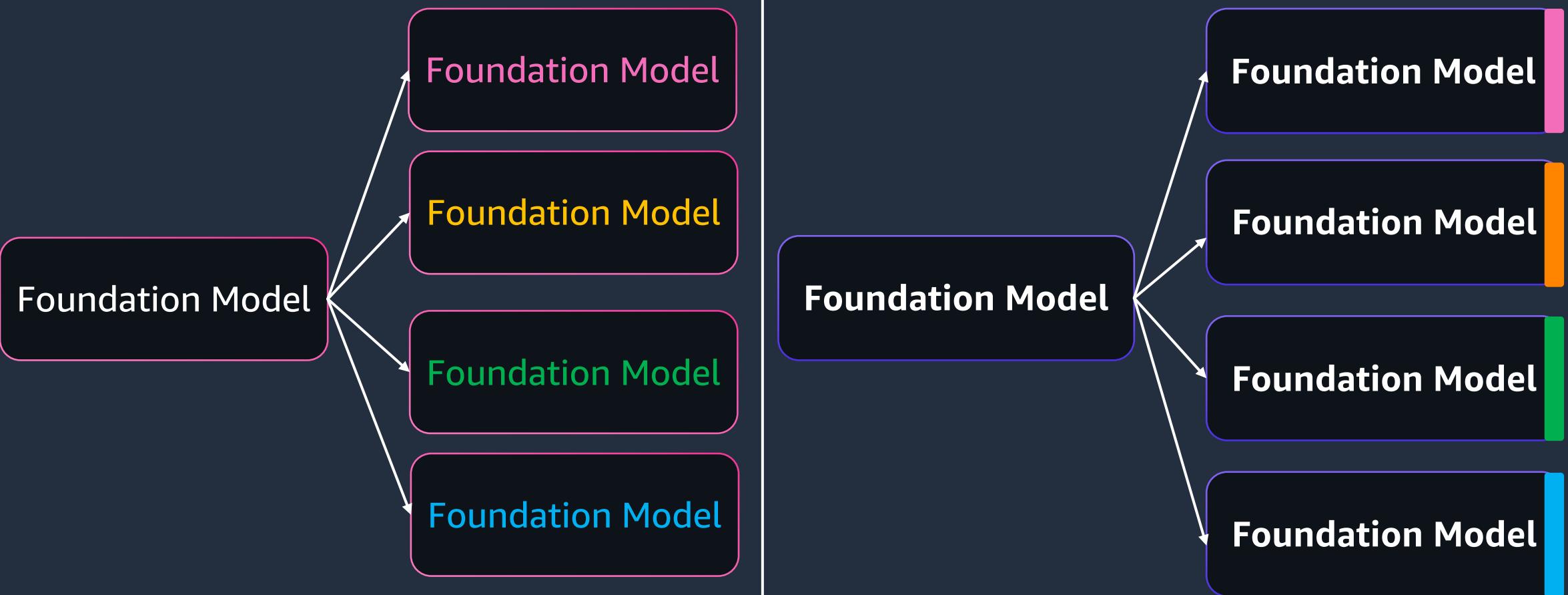
# Parameter-Efficient Fine-tuning (PEFT)?

Normal Fine-tuning

$h$

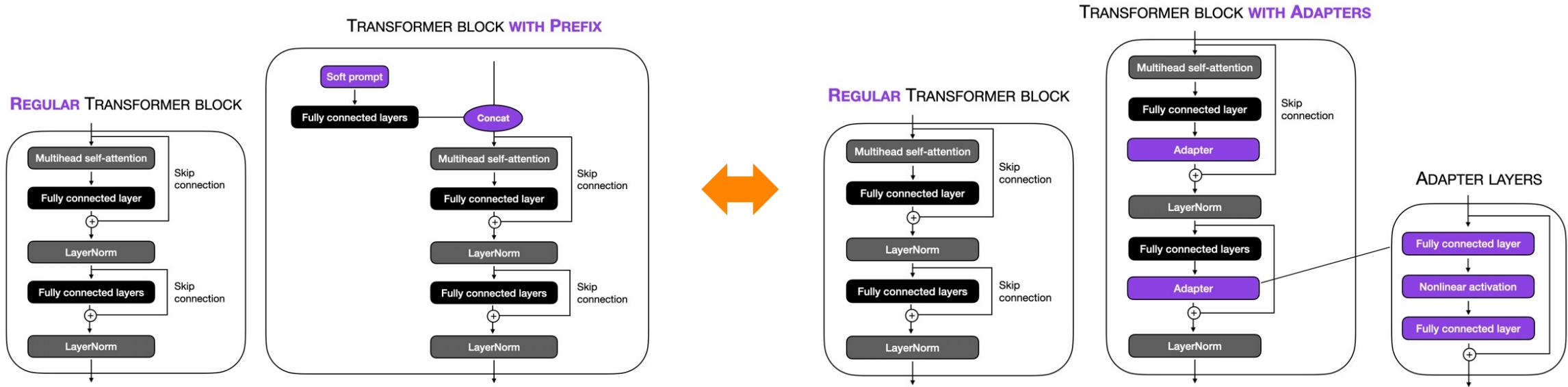
PEFT

$h' = h + \Delta h$



# PEFT Techniques – Adapters

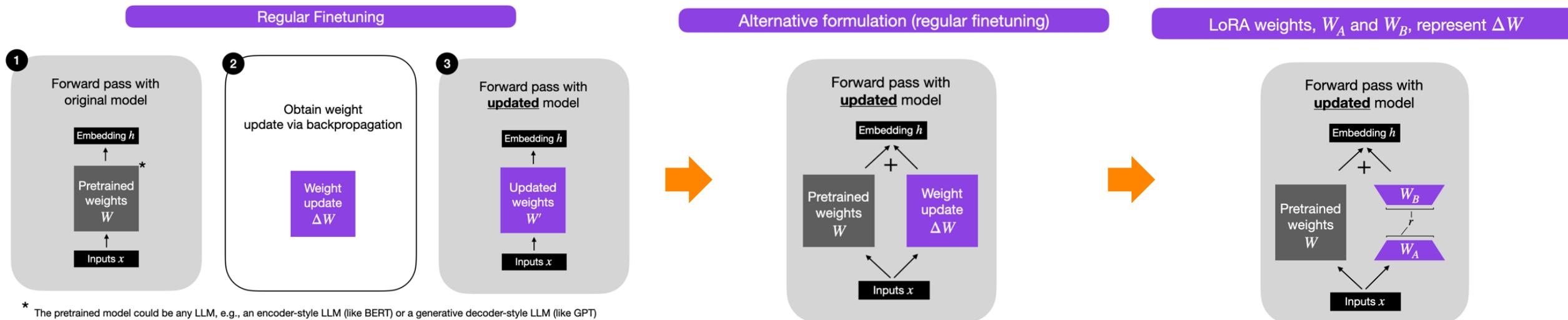
Understanding Parameter-Efficient LLM Finetuning: Prompt Tuning and Prefix Tuning



- 어댑터 튜닝: 트랜스포머 레이어 사이에 병목 레이어 (= 어댑터)를 추가하고 훈련함

# PEFT Techniques – LoRA

Parameter-Efficient LLM Finetuning with Low-Rank Adaptation (LoRA)

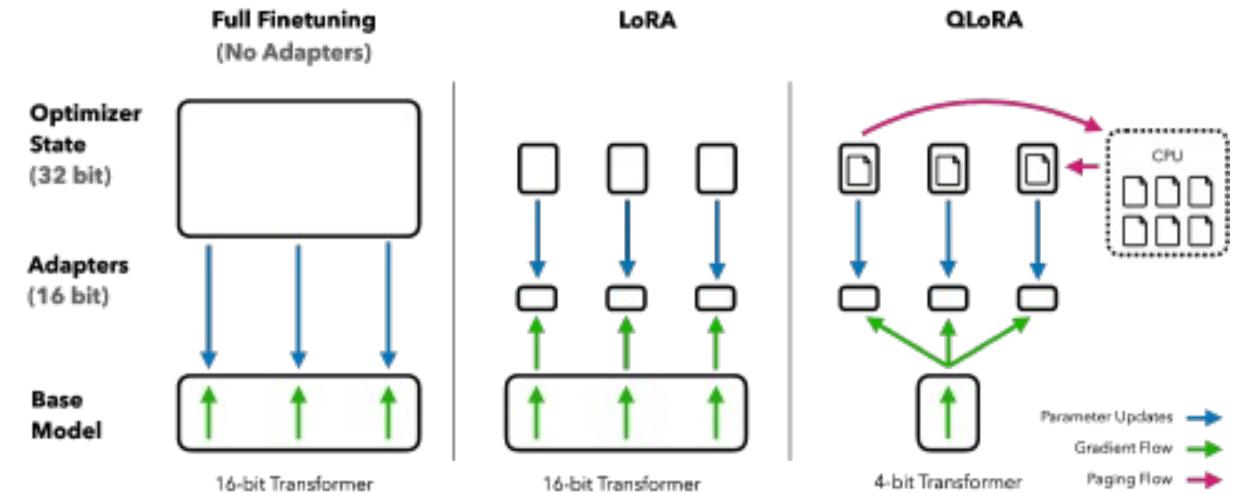
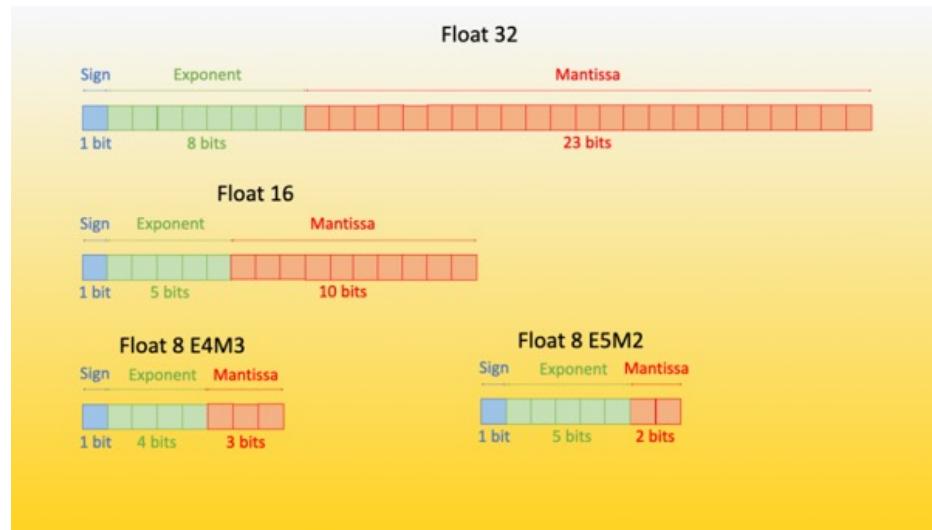


$$\begin{matrix} & 2,000 \text{ columns} \\ \begin{matrix} 1,000 \text{ rows} \\ \text{Matrix A} \end{matrix} & = \end{matrix} \begin{matrix} 2 \text{ columns} \\ \begin{matrix} 1,000 \text{ rows} \\ \text{Matrix B} \end{matrix} \end{matrix} \cdot \begin{matrix} 2,000 \text{ columns} \\ \begin{matrix} 2 \text{ rows} \\ \text{Matrix C} \end{matrix} \end{matrix}$$

- 기존 모델 가중치는 고정시키고 추가로 더해주는 정도를 학습함
- 가중치 전체 대신 표현의 일부만 학습 (저차원 행렬)
- 범용성 높아 이미지 도메인에도 적용 가능

# PEFT Techniques – QLoRA

[A Gentle Introduction to 8-bit Matrix Multiplication for Transformers at Scale Using Hugging Face Transformers, Accelerate and Bitsandbytes](#)



- LoRA 가중치에 4비트 양자화 (경량화) 적용
- 가중치를 4비트 NormalFloat 자료형으로 저장하되 모델 학습에서 필요한 경우 bfloat16으로 복원시켜서 사용
- 16비트 전체 파인튜닝과 성능 거의 동일하되, 필요한 GPU 메모리 크기 현저히 감소

# Types of PEFT Techniques and Their Performance

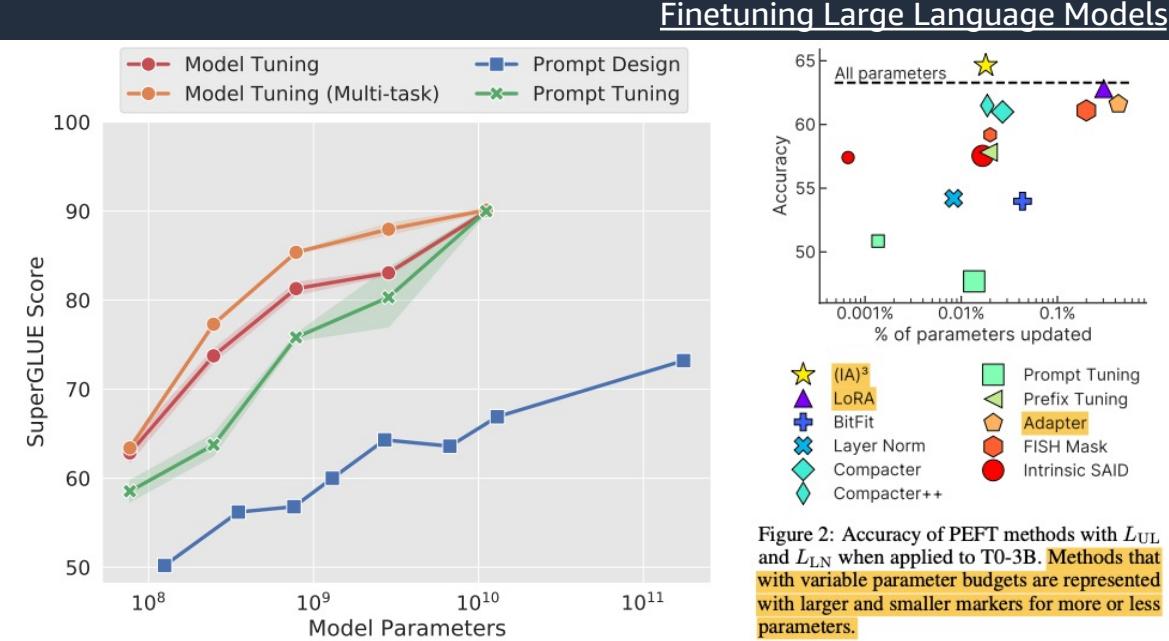
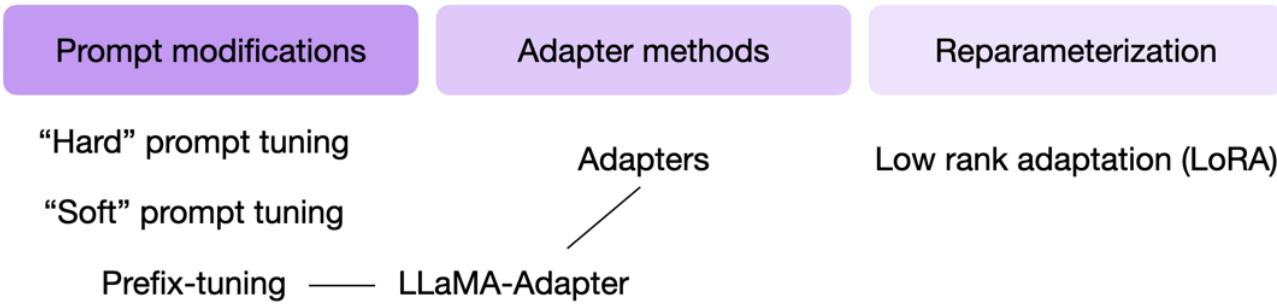


Figure 2: Accuracy of PEFT methods with  $L_{UL}$  and  $L_{LN}$  when applied to T0-3B. Methods that with variable parameter budgets are represented with larger and smaller markers for more or less parameters.

|    | <b>Prompt Tuning</b>                                                                                                                  | <b>Prefix Tuning</b>                                                                       | <b>Adapter</b>                                                                                | <b>LoRA</b>                                                                                                        |
|----|---------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------|
| 장점 | <ul style="list-style-type: none"> <li>하드 프롬프트 튜닝보다 높은 성능</li> </ul>                                                                  | <ul style="list-style-type: none"> <li>전체 파인튜닝에 준하는 성능</li> </ul>                          | <ul style="list-style-type: none"> <li>콘텍스트 길이 감소 없음</li> </ul>                               | <ul style="list-style-type: none"> <li>전체 파인튜닝에 준하는 성능</li> <li>콘텍스트 길이 감소 없음</li> <li>추론 시 추가 연산 거의 없음</li> </ul> |
| 단점 | <ul style="list-style-type: none"> <li>전체 파인튜닝보다 성능 안좋지만 모델 크기 커질수록 균접해짐</li> <li>임베딩 벡터에 튜닝 가능한 텐서 도입으로 처리 가능한 콘텍스트 길이 감소</li> </ul> | <ul style="list-style-type: none"> <li>임베딩 벡터에 튜닝 가능한 텐서 도입으로 처리 가능한 콘텍스트 길이 감소</li> </ul> | <ul style="list-style-type: none"> <li>프리픽스 튜닝보다 약간 떨어지는 성능</li> <li>추론 시 추가 연산 발생</li> </ul> |                                                                                                                    |

# Implementing PEFT on SageMaker

- HuggingFace에서 주요 PEFT 기법 지원
- Examples 을 통한 SageMaker에서 수행 가능
- 대규모 모델을 위한 😊 Accelerate와 통합되어 DeepSpeed 및 LM 추론 활용
- int8 양자화(예: LoRA + int8 양자화)를 쉽게 연동 가능

- LoRA
- Prefix Tuning
- P-Tuning
- Prompt Tuning
- AdaLoRA
- $(IA)^3$

```
from transformers import AutoModelForSeq2SeqLM
from peft import get_peft_config, get_peft_model, LoraConfig, TaskType
model_name_or_path = "bigscience/mt0-large"
tokenizer_name_or_path = "bigscience/mt0-large"

peft_config = LoraConfig(
    task_type=TaskType.SEQ_2_SEQ_LM, inference_mode=False, r=8, lora_alpha=32, lora_dropout=0.1
)

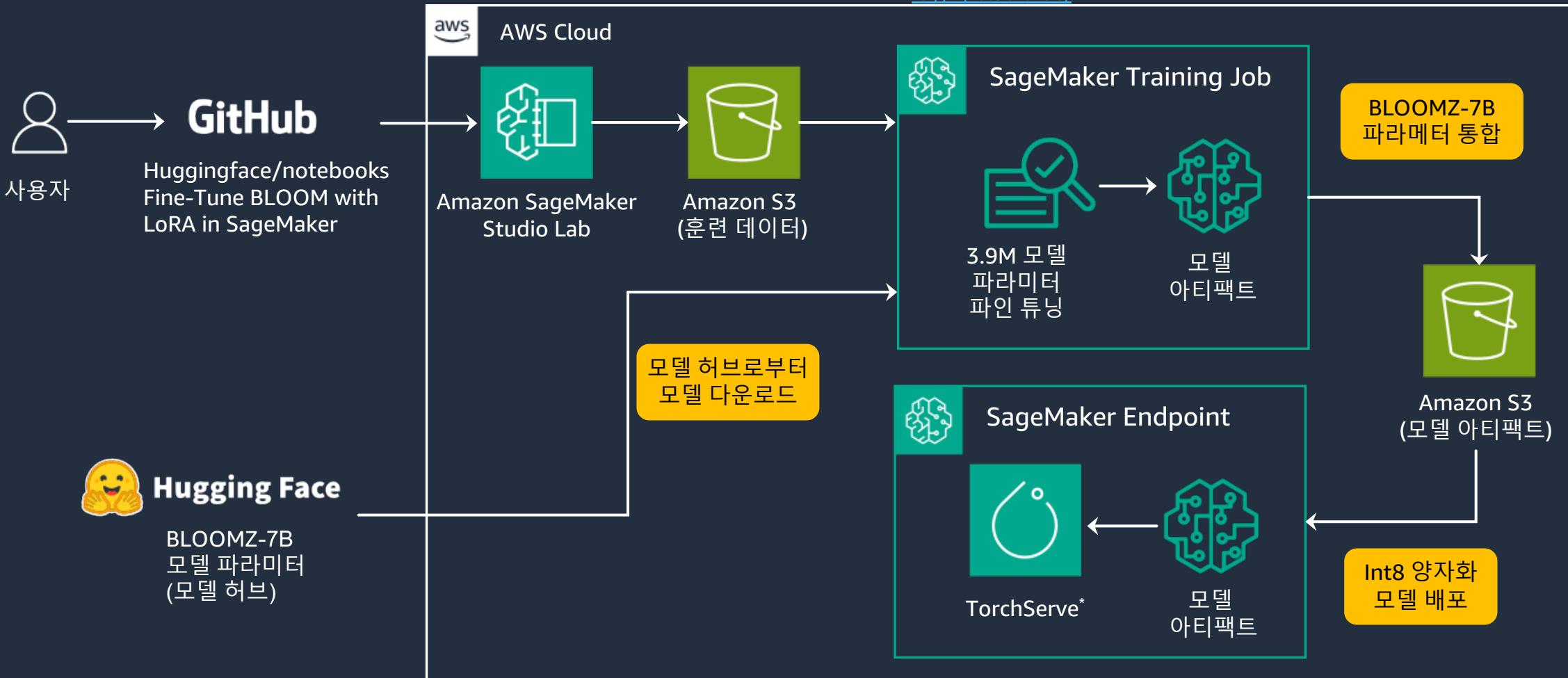
model = AutoModelForSeq2SeqLM.from_pretrained(model_name_or_path)
model = get_peft_model(model, peft_config)
model.print_trainable_parameters()
# output: trainable params: 2359296 || all params: 1231940608 || trainable%: 0.19151053100118282
```

Ref : <https://github.com/huggingface/peft>  
[https://github.com/huggingface/notebooks/blob/main/sagemaker/24\\_train\\_bloom\\_peft\\_lora/sagemaker-notebook.ipynb](https://github.com/huggingface/notebooks/blob/main/sagemaker/24_train_bloom_peft_lora/sagemaker-notebook.ipynb)  
[https://github.com/aws-samples/aws-ai-ml-workshop-kr/tree/master/genai/jumpstart/text\\_to\\_text](https://github.com/aws-samples/aws-ai-ml-workshop-kr/tree/master/genai/jumpstart/text_to_text)

# PEFT 사례: BLOOMZ-7B 파인 투닝 및 배포

AWS 기술 블로그  
허깅페이스와 LoRA를 사용하여 단일 Amazon SageMaker GPU에서 대규모 언어 모델(LLM) 훈련하기  
by Daekeun Kim and Hyeongsang Jeon | on 17 7월 2023 | in Advanced (300), Amazon SageMaker, Artificial Intelligence, Generative AI, Technical How-To | Permalink | Share

Blog: <https://aws.amazon.com/ko/blogs/tech/train-a-large-language-model-on-a-single-amazon-sagemaker-gpu-hugging-face-and-lora/>



핸즈온: <https://github.com/daekeun-ml/sm-distributed-train-bloom-peft-lora>

Falcon-40B PEFT 사례: <https://aws.amazon.com/blogs/machine-learning/interactively-fine-tune-falcon-40b-and-other-llms-on-amazon-sagemaker-studio-notebooks-using-qlora-aws>



# Thank you!

Hyo  
Solutions Architect