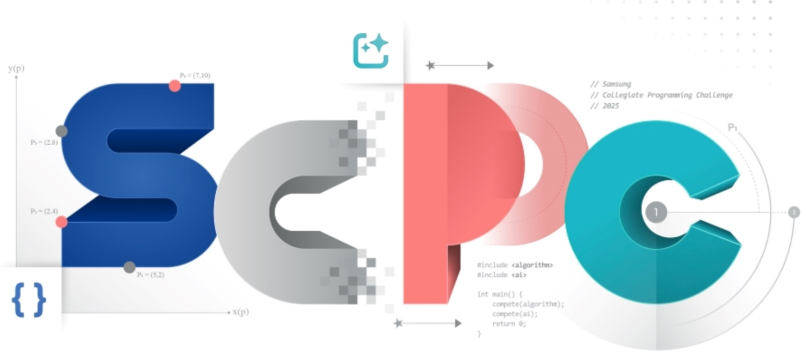
Samsung Collegiate Programming Challenge 2025

AI

모델링 전략, 데이터처리, 성능 최적화를 통한
생성형 AI 모델 개발

참가자 손희경



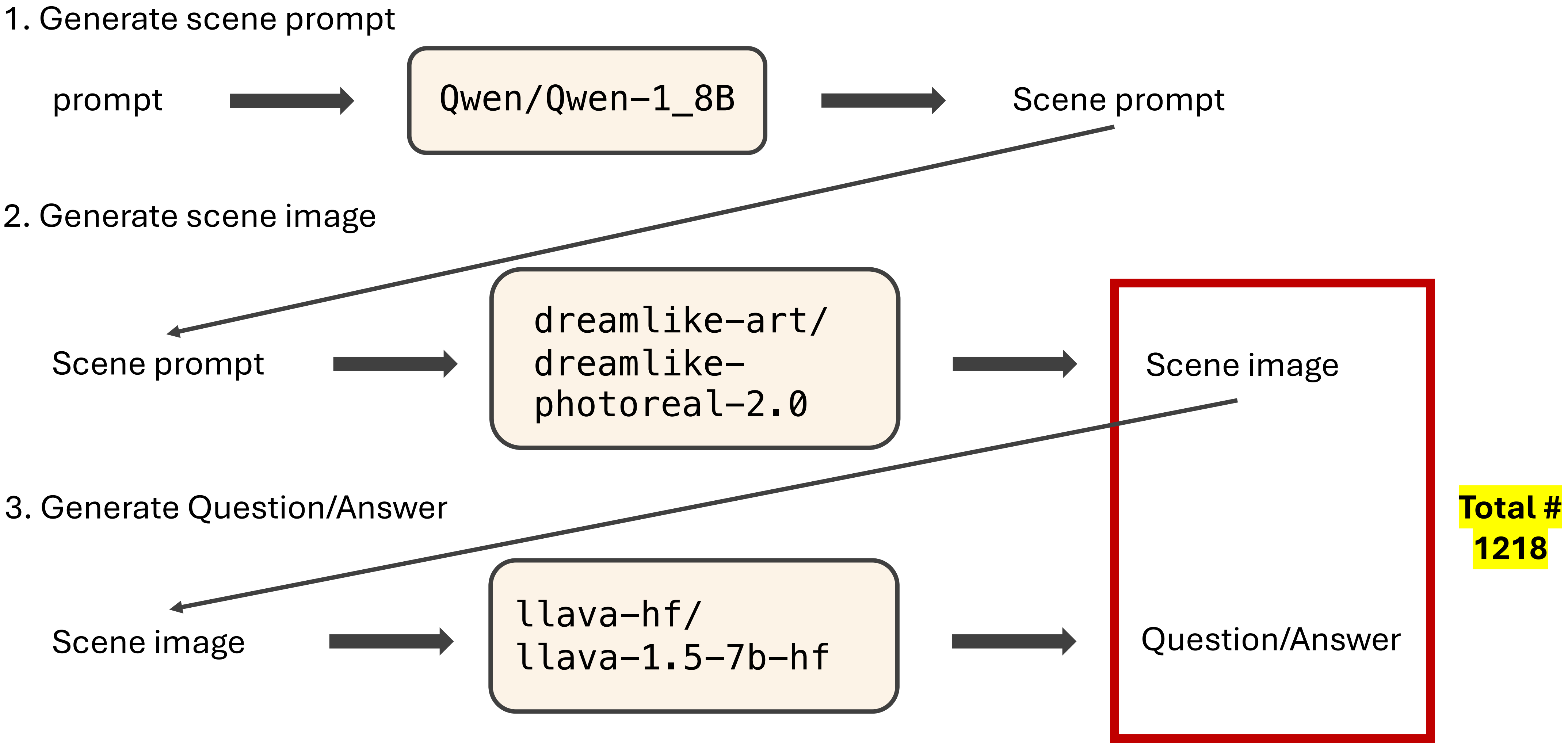
```
// Samsung
// Collegiate Programming Challenge
// 2025
```

```
#include <algorithm>
#include <ai>

int main() {
    compete(algorithm);
    compete(ai);
    return 0;
}
```

Contents

- Generate train dataset
- BLIP2-flan-t5
- Partial quantization
- Finetune with LoRA Adapter
- Inference
- Evaluation

Generate train dataset



1. Generate scene prompt

prompt → Qwen/Qwen-1_8B → Scene prompt

2. Generate scene image

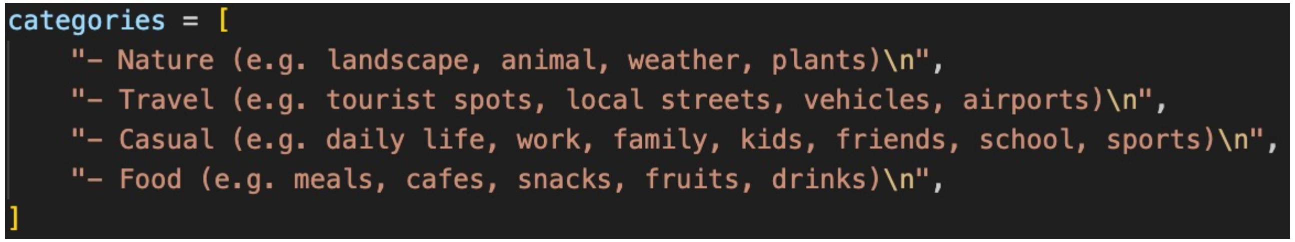
Scene prompt → dreamlike-art/dreamlike-photoreal-2.0 → Scene image

3. Generate Question/Answer

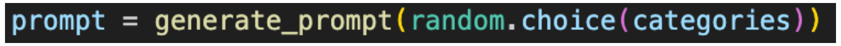
Scene image → llava-hf/llava-1.5-7b-hf → Question/Answer

Total # 1218

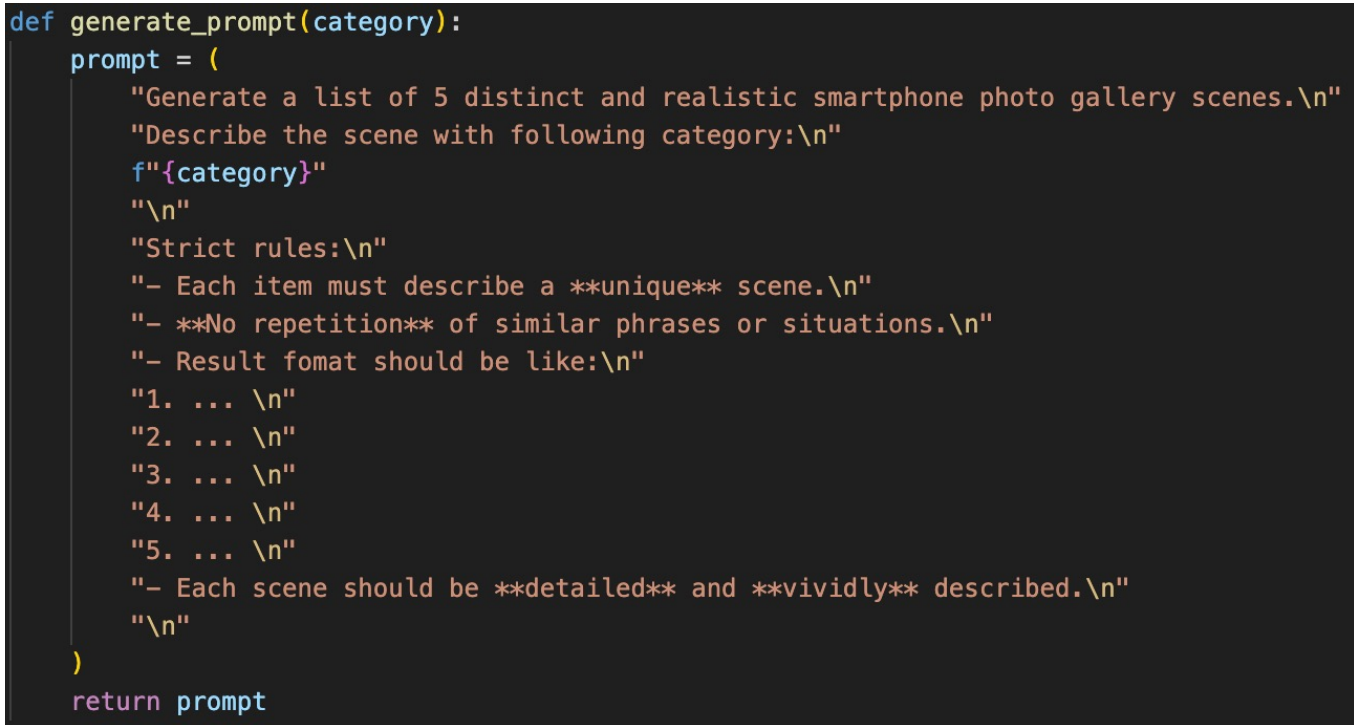
1. Generate scene prompt



```python
categories = [
    "- Nature (e.g. landscape, animal, weather, plants)\n",
    "- Travel (e.g. tourist spots, local streets, vehicles, airports)\n",
    "- Casual (e.g. daily life, work, family, kids, friends, school, sports)\n",
    "- Food (e.g. meals, cafes, snacks, fruits, drinks)\n",
]
```



```python
prompt = generate_prompt(random.choice(categories))
```



```python
def generate_prompt(category):
    prompt = (
        "Generate a list of 5 distinct and realistic smartphone photo gallery scenes.\n"
        "Describe the scene with following category:\n"
        f"{category}"
        "\n"
        "Strict rules:\n"
        "- Each item must describe a **unique** scene.\n"
        "- **No repetition** of similar phrases or situations.\n"
        "- Result fomat should be like:\n"
        "1. ... \n"
        "2. ... \n"
        "3. ... \n"
        "4. ... \n"
        "5. ... \n"
        "- Each scene should be **detailed** and **vividly** described.\n"
        "\n"
    )
    return prompt
```

example

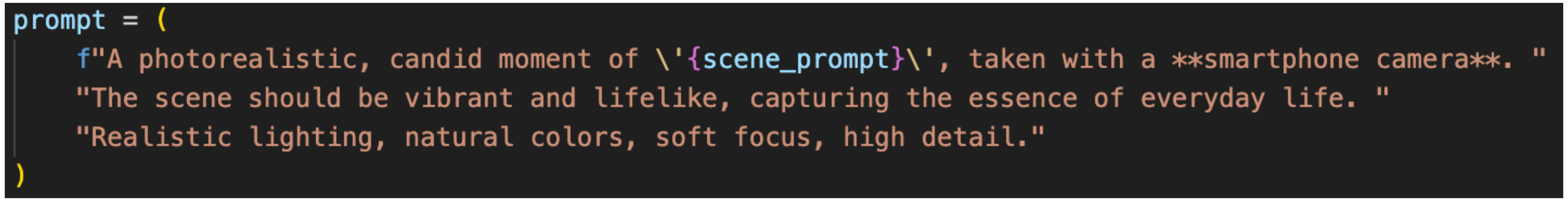
A group of friends sharing a picnic at the beach, enjoying a delicious seafood platter and enjoying a relaxing afternoon on the sand.

A group of friends gathered around a table, surrounded by an array of colorful plates of sushi, sashimi, and other Japanese cuisine.

I took this picture at the popular tourist spot in Paris, France. The streets were bustling with people and cars, the air was fresh and the smells of coffee and croissants were wafting through the air. The iconic Eiffel Tower was perched on the edge of the city, casting long shadows across the bustling city.

A picture of a cute puppy lounging on the grass with a bag of dog treats.

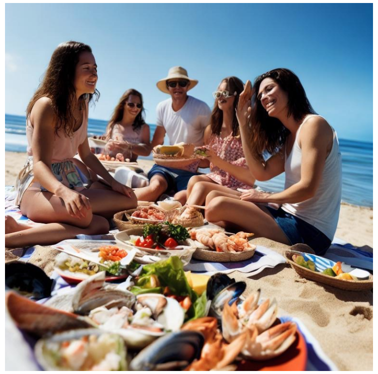
2. Generate scene image



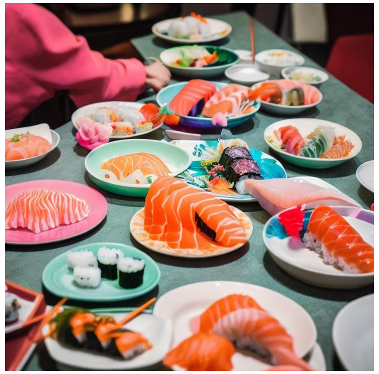
```
prompt = (
    f"A photorealistic, candid moment of \'{scene_prompt}\', taken with a **smartphone camera**. "
    "The scene should be vibrant and lifelike, capturing the essence of everyday life. "
    "Realistic lighting, natural colors, soft focus, high detail."
)
```

example

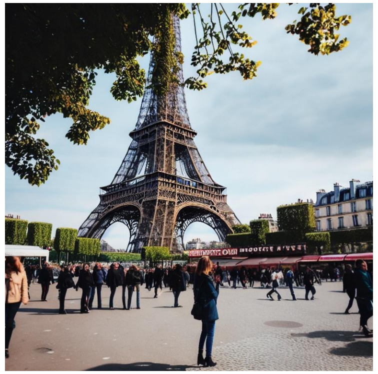
A group of friends sharing a picnic at the beach, enjoying a delicious seafood platter and enjoying a relaxing afternoon on the sand.



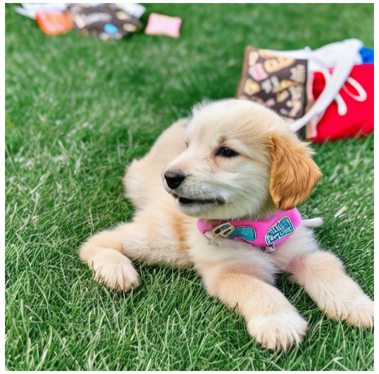
A group of friends gathered around a table, surrounded by an array of colorful plates of sushi, sashimi, and other Japanese cuisine.



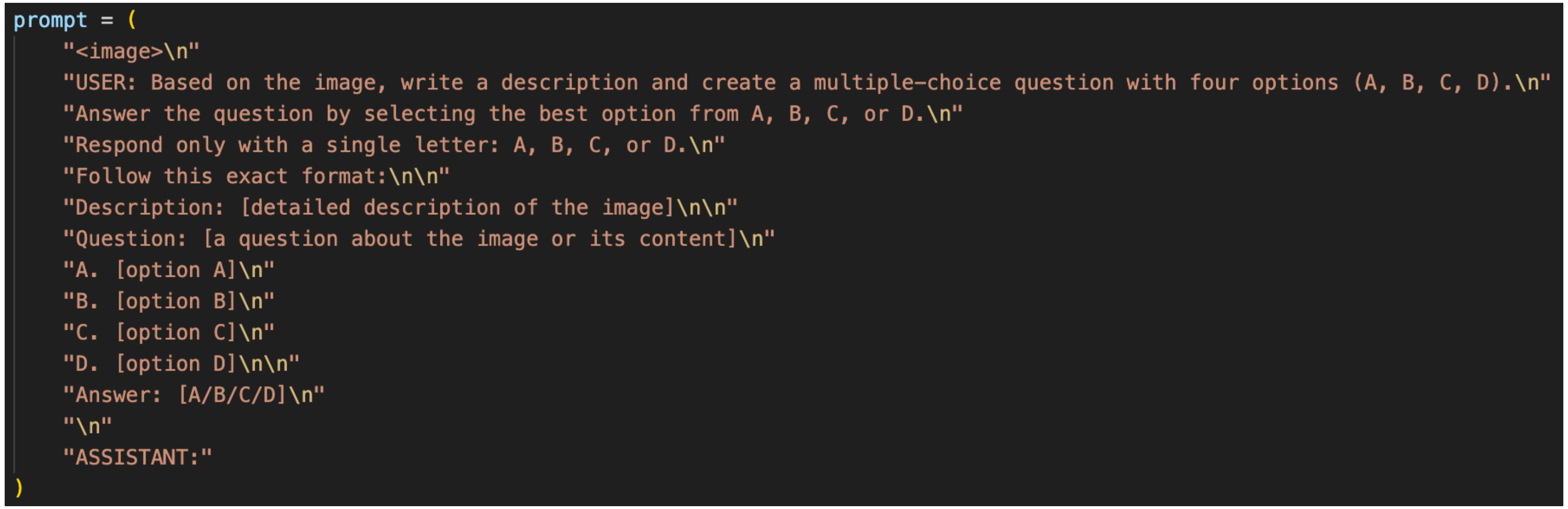
I took this picture at the popular tourist spot in Paris, France. The streets were bustling with people and cars, the air was fresh and the smells of coffee and croissants were wafting through the air. The iconic Eiffel Tower was perched on the edge of the city, casting long shadows across the bustling city.



A picture of a cute puppy lounging on the grass with a bag of dog treats.

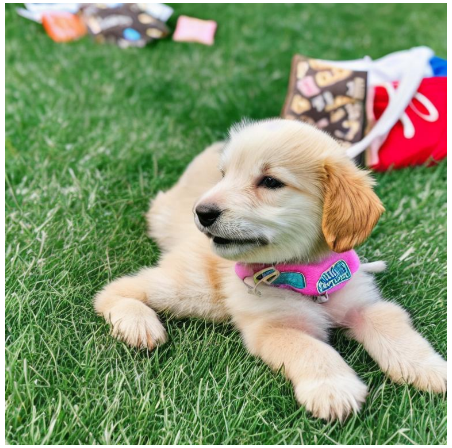


3. Generate Question/Answer



```
prompt = (
    "<image>\n"
    "USER: Based on the image, write a description and create a multiple-choice question with four options (A, B, C, D).\n"
    "Answer the question by selecting the best option from A, B, C, or D.\n"
    "Respond only with a single letter: A, B, C, or D.\n"
    "Follow this exact format:\n\n"
    "Description: [detailed description of the image]\n\n"
    "Question: [a question about the image or its content]\n"
    "A. [option A]\n"
    "B. [option B]\n"
    "C. [option C]\n"
    "D. [option D]\n\n"
    "Answer: [A/B/C/D]\n"
    "\n"
    "ASSISTANT:"
)
```

example



Description

A small brown and white dog is laying on the grass. It is wearing a pink collar and appears to be looking at the camera. There are also some bags and a backpack nearby.

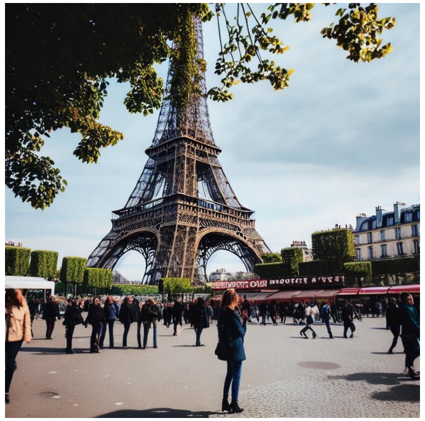
Question

What color is the dog's collar?

Option & Answer

Pink	Black	Blue	Green	A

More examples



Description

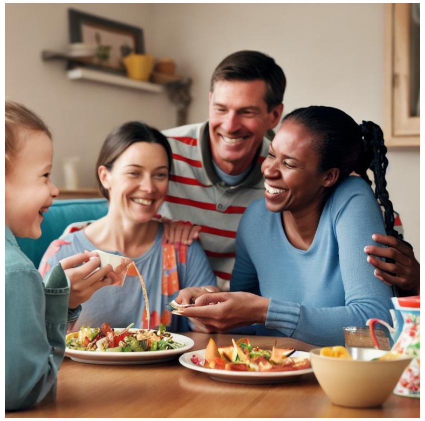
A large crowd of people is gathered in front of the Eiffel Tower, which is a famous landmark in Paris. The tower is surrounded by trees and buildings, creating a picturesque scene.

Question

What is the main attraction in the image?

Option & Answer

The Eiffel Tower	The Arc de Triomphe	The Louvre Museum	Notre-Dame Cathedral	A



Description

A family of four, including a man, a woman, and two children, are sitting around a wooden dining table, enjoying a meal together. They are smiling and laughing, creating a warm and happy atmosphere.

Question

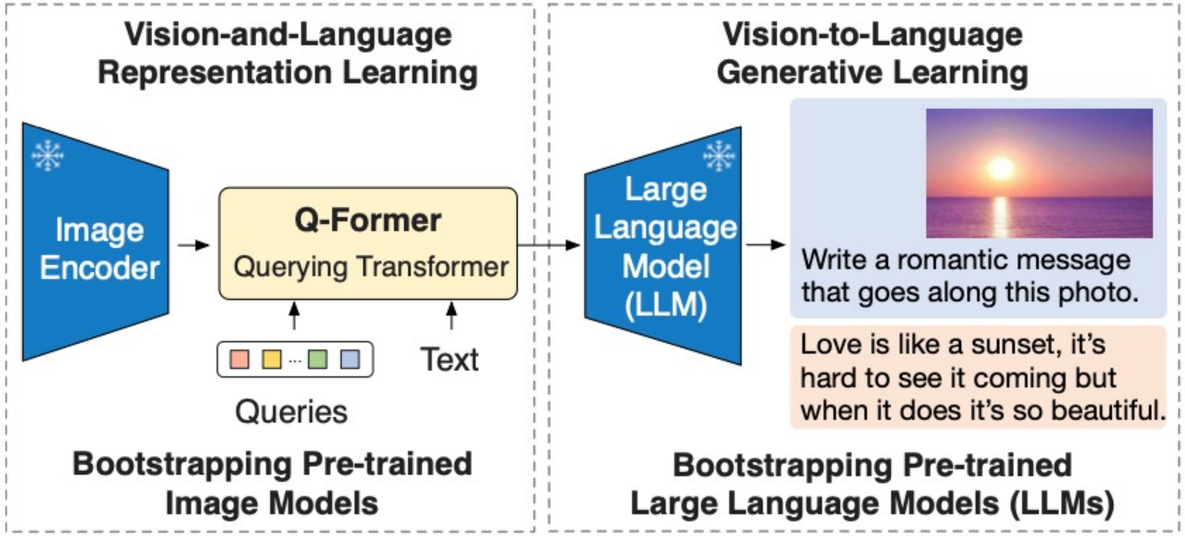
What is the family doing in the image?

Option & Answer

They are eating pizza	They are having a meal together	They are playing a game	They are watching TV	B

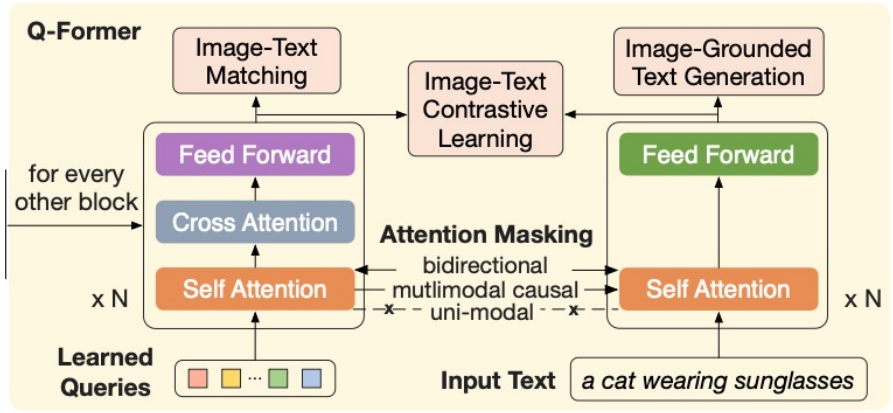
BLIP2-flan-t5

Overview of BLIP-2's framework



Write a romantic message that goes along this photo.

Love is like a sunset, it's hard to see it coming but when it does it's so beautiful.



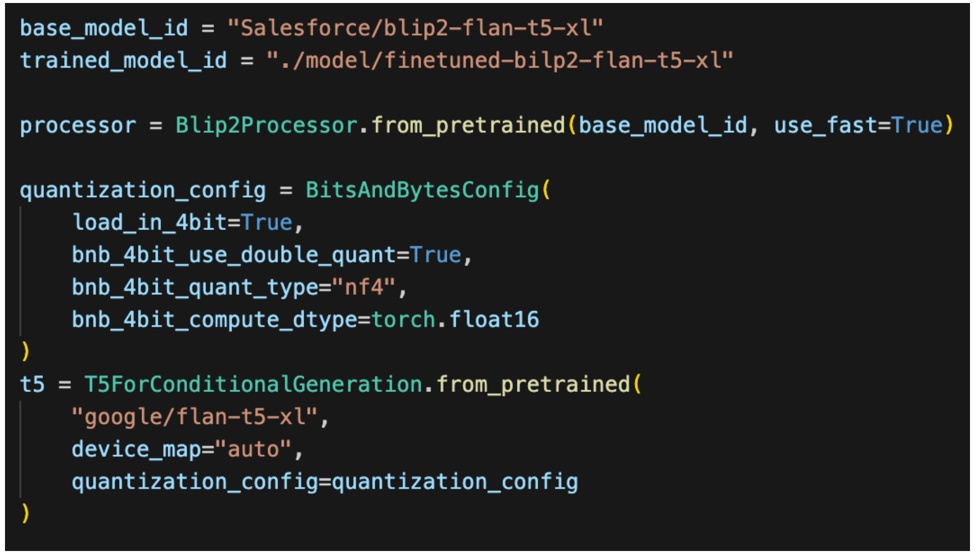
Salesforce/bilp2-flan-t5-xl

- BLIP-2 아키텍처에 FLAN-T5-XL 텍스트 디코더를 결합한 멀티모달 모델
- FLAN-T5 기반으로 Instruction-following 능력이 우수
- 다양한 태스크 지원 (설명, 캡셔닝, 질문 생성, 추론 등)

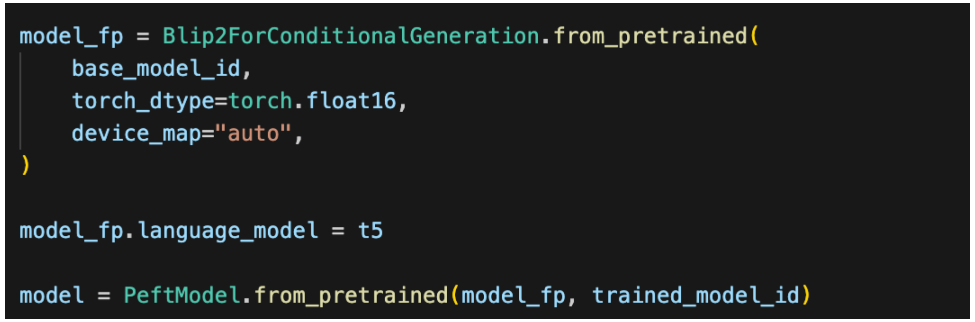
Partial quantization

Strategy

- T5 decoder만 4bit quantization하여 loading
- Vision encoder, Q-Former는 그대로



```
base_model_id = "Salesforce/blip2-flan-t5-xl"
trained_model_id = "./model/finetuned-bilp2-flan-t5-xl"

processor = Blip2Processor.from_pretrained(base_model_id, use_fast=True)

quantization_config = BitsAndBytesConfig(
    load_in_4bit=True,
    bnb_4bit_use_double_quant=True,
    bnb_4bit_quant_type="nf4",
    bnb_4bit_compute_dtype=torch.float16
)
t5 = T5ForConditionalGeneration.from_pretrained(
    "google/flan-t5-xl",
    device_map="auto",
    quantization_config=quantization_config
)
```



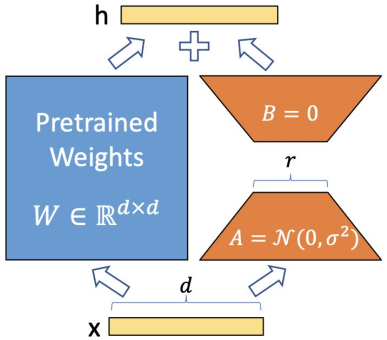
```
model_fp = Blip2ForConditionalGeneration.from_pretrained(
    base_model_id,
    torch_dtype=torch.float16,
    device_map="auto",
)

model_fp.language_model = t5

model = PeftModel.from_pretrained(model_fp, trained_model_id)
```

Result of parameter reduction

- `Before: 3,949,180,416`
- `After: 2,841,884,160`

Finetune with LoRA Adapter

LoRA (Low-Rank Adaptation)

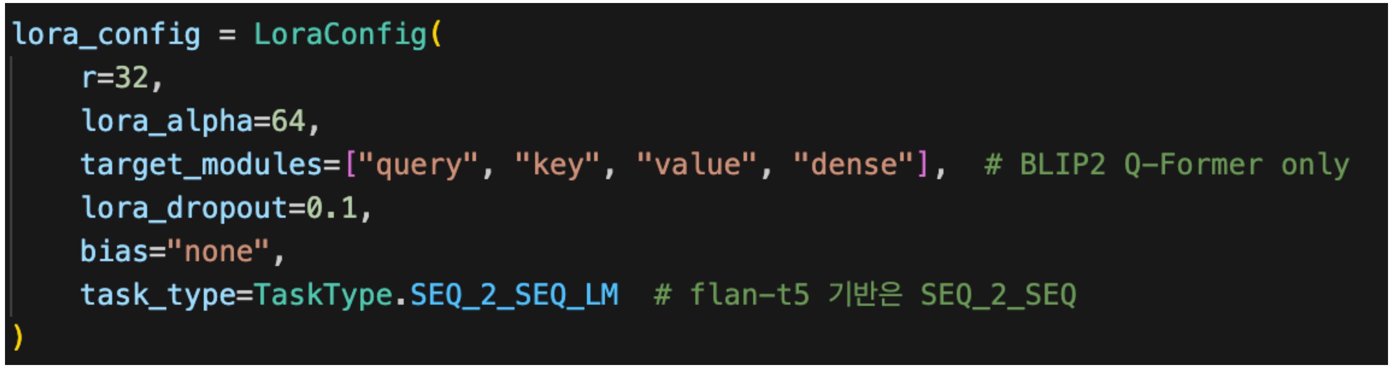

h
Pretrained Weights
$W \in \mathbb{R}^{d\times d}$
$B = 0$
r
$A = \mathcal{N}(0, \sigma^2)$
d
x

$$h = W_0x + BAx$$

- HuggingFace에서 개발한 Parameter-Efficient Fine-Tuning (PEFT) 방식 중 하나
- 사전 훈련된 모델 가중치는 고정하고 훈련 가능한 레이어(LoRA adapter의 A Layer, B Layer) 들을 별도로 붙이고 추가 훈련을 통해 학습
- 장점: Memory Efficient, Task Adaptability, Composable / Pluggable

Lora Configuration

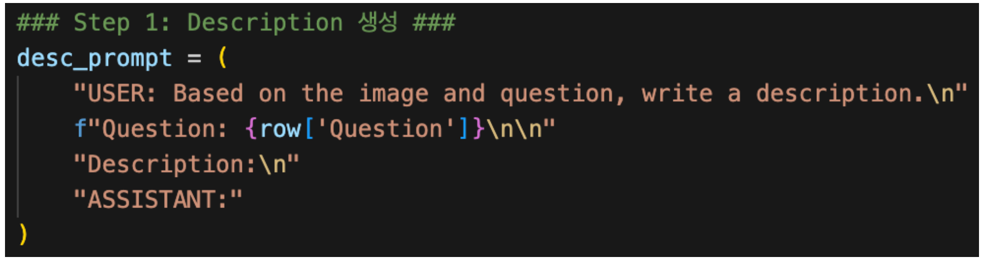
- Q-Fromer module만 target으로 지정



```
lora_config = LoraConfig(
    r=32,
    lora_alpha=64,
    target_modules=["query", "key", "value", "dense"],  # BLIP2 Q-Former only
    lora_dropout=0.1,
    bias="none",
    task_type=TaskType.SEQ_2_SEQ_LM  # flan-t5 기반은 SEQ_2_SEQ
)
```

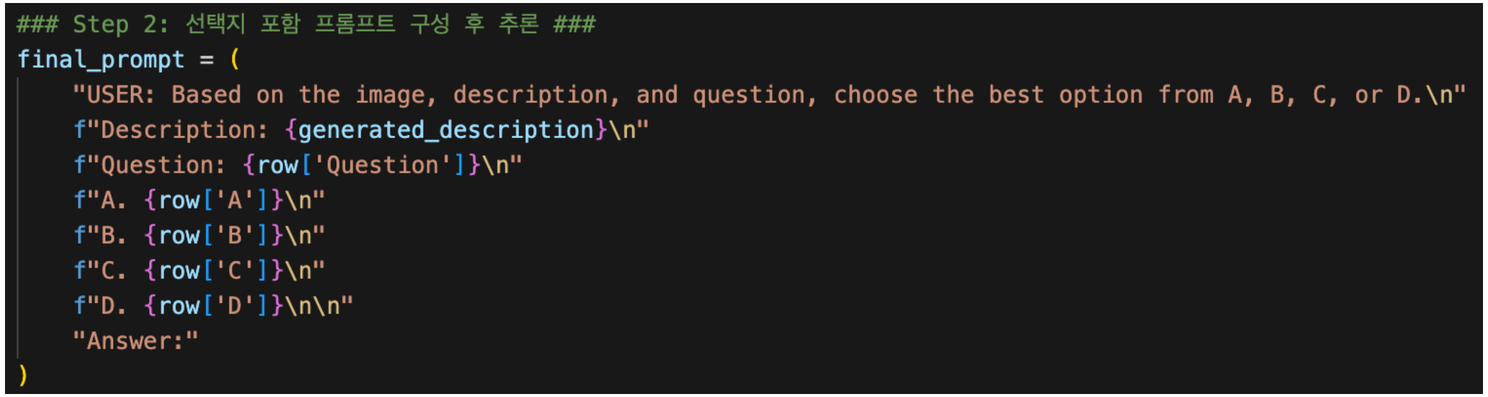
Inference

2-stage Inference Strategy



```
### Step 1: Description 생성 ###
desc_prompt = (
    "USER: Based on the image and question, write a description.\n"
    f"Question: {row['Question']}\n\n"
    "Description:\n"
    "ASSISTANT:"
)
```

```
[Step 1] Generated Description:
a pot of dumplings being cooked in a steamer
```

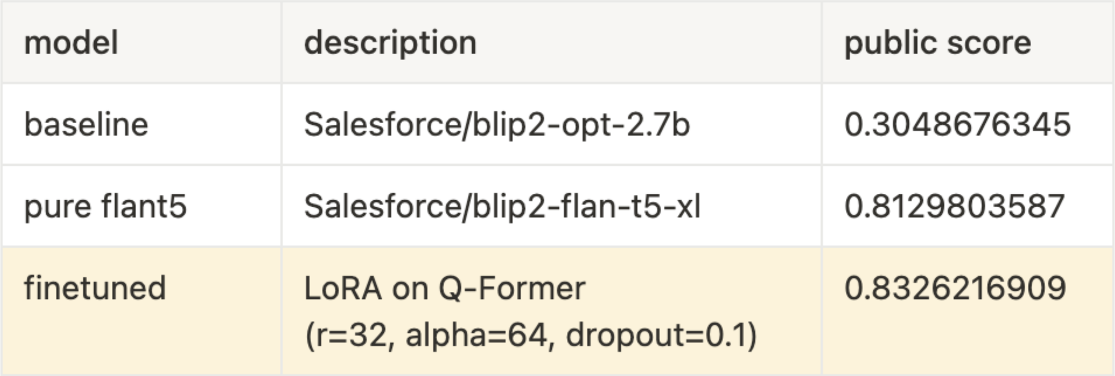


```
### Step 2: 선택지 포함 프롬프트 구성 후 추론 ###
final_prompt = (
    "USER: Based on the image, description, and question, choose the best option from A, B, C, or D.\n"
    f"Description: {generated_description}\n"
    f"Question: {row['Question']}\n"
    f"A. {row['A']}\n"
    f"B. {row['B']}\n"
    f"C. {row['C']}\n"
    f"D. {row['D']}\n\n"
    "Answer:"
)
```

```
[Step 2] Final Answer Prediction: A
```

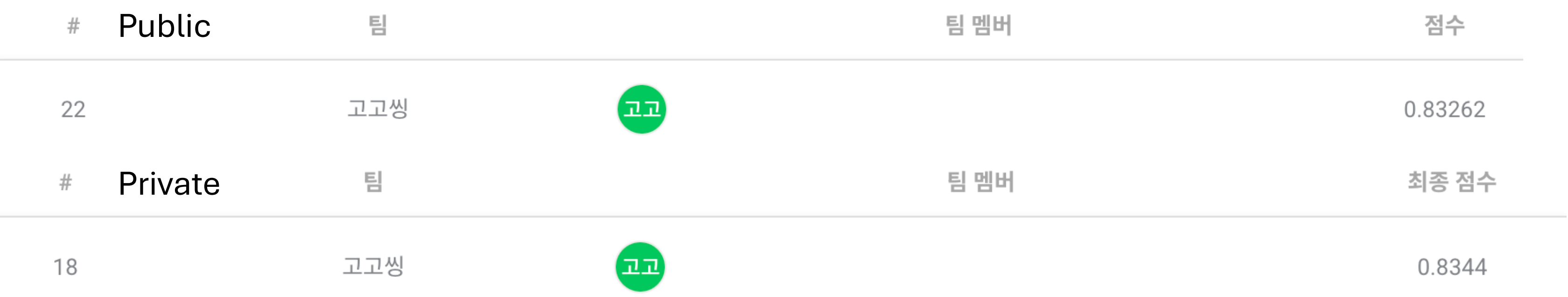
Evaluation

Evaluation result



model	description	public score
baseline	Salesforce/blip2-opt-2.7b	0.3048676345
pure flant5	Salesforce/blip2-flan-t5-xl	0.8129803587
finetuned	LoRA on Q-Former (r=32, alpha=64, dropout=0.1)	0.8326216909

Final Public, Private score



Public

#	팀	팀 멤버	점수
22	고고씽	고고	0.83262

Private

#	팀	팀 멤버	최종 점수
18	고고씽	고고	0.8344