

# VQA 스터디

김태경 | 박찬욱

# Contents

01 스터디 활동 리뷰 (논문 분석)

02 스터디 활동 리뷰 (대회 출전)

03 향후 계획

04 질문 및 답변

# 01 스터디 활동 리뷰 (논문 분석)

BLIP: Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation

BLIP(Bootstrapping Language-Image Pre-training)는 이미지와 텍스트를 동시에 이해하고 BLIP(Bootstrapping Language-Image Pre-training)는 이미지와 텍스트를 동시에 이해하고 생성할 수 있도록 설계된 통합 비전-언어 모델이다.

이 논문은 웹 데이터를 활용한 효과적인 사전학습 방식을 제안하여, 이미지 캡셔닝, 비주얼 질의응답 등 다양한 비전-언어 과제에서 뛰어난 성능을 달성한다.

arXiv:2201.12086v2 [cs.CV] 15 Feb 2022

## BLIP: Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation

Junnan Li Dongxu Li Caiming Xiong Steven Hoi  
Salesforce Research  
<https://github.com/salesforce/BLIP>

### Abstract

Vision-Language Pre-training (VLP) has advanced the performance for many vision-language tasks. However, most existing pre-trained models only excel in either understanding-based tasks or generation-based tasks. Furthermore, performance improvement has been largely achieved by scaling up the dataset with noisy image-text pairs collected from the web, which is a suboptimal source of supervision. In this paper, we propose BLIP, a new VLP framework which transfers flexibly to both vision-language understanding and generation tasks. BLIP effectively utilizes the noisy web data by bootstrapping the captions, where a captioner generates synthetic captions and a filter removes the noisy ones. We achieve state-of-the-art results on a wide range of vision-language tasks, such as image-text retrieval (+2.7% in average recall@1), image captioning (+2.8% in CIDEr), and VQA (+1.6% in VQA score). BLIP also demonstrates strong generalization ability when directly transferred to video-language tasks in a zero-shot manner. Code, models, and datasets are released.

### 1. Introduction

Vision-language pre-training has recently received tremendous success on various multimodal downstream tasks. However, existing methods have two major limitations:

- (1) Model perspective: most methods either adopt an encoder-based model (Radford et al., 2021; Li et al., 2021a), or an encoder-decoder (Cho et al., 2021; Wang et al., 2021) model. However, encoder-based models are less straightforward to directly transfer to text generation tasks (e.g. image captioning), whereas encoder-decoder models have not been successfully adopted for image-text retrieval tasks.
- (2) Data perspective: most state-of-the-art methods (e.g., CLIP (Radford et al., 2021), ALBEF (Li et al., 2021a), SimVLM (Wang et al., 2021)) pre-train on image-text pairs

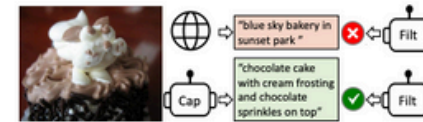


Figure 1. We use a Captioner (Cap) to generate synthetic captions for web images, and a Filter (Filt) to remove noisy captions.

collected from the web. Despite the performance gain obtained by scaling up the dataset, our paper shows that the noisy web text is suboptimal for vision-language learning.

To this end, we propose BLIP: Bootstrapping Language-Image Pre-training for unified vision-language understanding and generation. BLIP is a new VLP framework which enables a wider range of downstream tasks than existing methods. It introduces two contributions from the model and data perspective, respectively:

(a) Multimodal mixture of Encoder-Decoder (MED): a new model architecture for effective multi-task pre-training and flexible transfer learning. An MED can operate either as a unimodal encoder, or an image-grounded text encoder, or an image-grounded text decoder. The model is jointly pre-trained with three vision-language objectives: image-text contrastive learning, image-text matching, and image-conditioned language modeling.

(b) Captioning and Filtering (CapFilt): a new dataset bootstrapping method for learning from noisy image-text pairs. We finetune a pre-trained MED into two modules: a *captioner* to produce synthetic captions given web images, and a *filter* to remove noisy captions from both the original web texts and the synthetic texts.

We perform extensive experiments and analysis, and make the following key observations.

- We show that the captioner and the filter work together to achieve substantial performance improvement on various downstream tasks by bootstrapping the captions. We also find that more diverse captions yield larger gains.
- BLIP achieves state-of-the-art performance on a wide range of vision-language tasks, including image-text re-

### Summary

현존하는 VLP model은 understanding-based tasks나 generation-based tasks  $\times$  1. 항상 인터넷에서 수집한 noisy image-text pairs를 가진 데이터셋을 커우는 것으로 이루어져 suboptimal하다. 논문은 vision-language understanding와 generation tasks 모두 유연하게 transfer하는 BLIP을 제안한다. BLIP은 caption을 bootstrapping함으로써 noisy web data를 효과적으로 활용한다. 즉, captioner가 synthetic captions을 생성하고 filter가 noisy한 것들을 제거한다. image-text retrieval, image captioning, VQA 등 넓은 범위의 vision-language tasks에서 SOTA를 달성한다. 또 zero-shot manner로 video-language tasks에 직접 transfer되었을 때 강한 일반화 능력을 입증한다.

BLIP은 모델과 데이터라는 2가지 측면에서 기여를 한다.

- Multimodal mixture of Encoder-Decoder (MED)

효과적인 multi-task pre-training과 flexible transfer learning을 위한 새로운 model architecture. MED는 unimodal encoder나 image-grounded text encoder나 image-grounded text decoder로도 작동할 수 있다. 모델은 3가지 vision-language objectives, 즉 image-text contrastive learning, image-text matching, image-conditioned language modeling에 공동으로 pre-train된다.

- Captioning and Filtering (CapFilt)

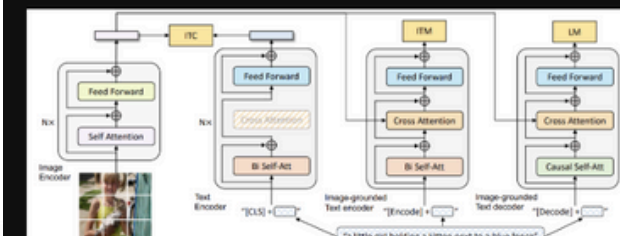
noisy image-text pairs로부터 학습하기 위한 새로운 dataset bootstrapping method. pre-trained MED를 2 모듈로, 즉 web images가 주어졌을 때 synthetic captions을 생성하기 위한 captioner와 original/synthetic web texts에서 noisy captions을 제거하기 위한 filter로 finetune된다.



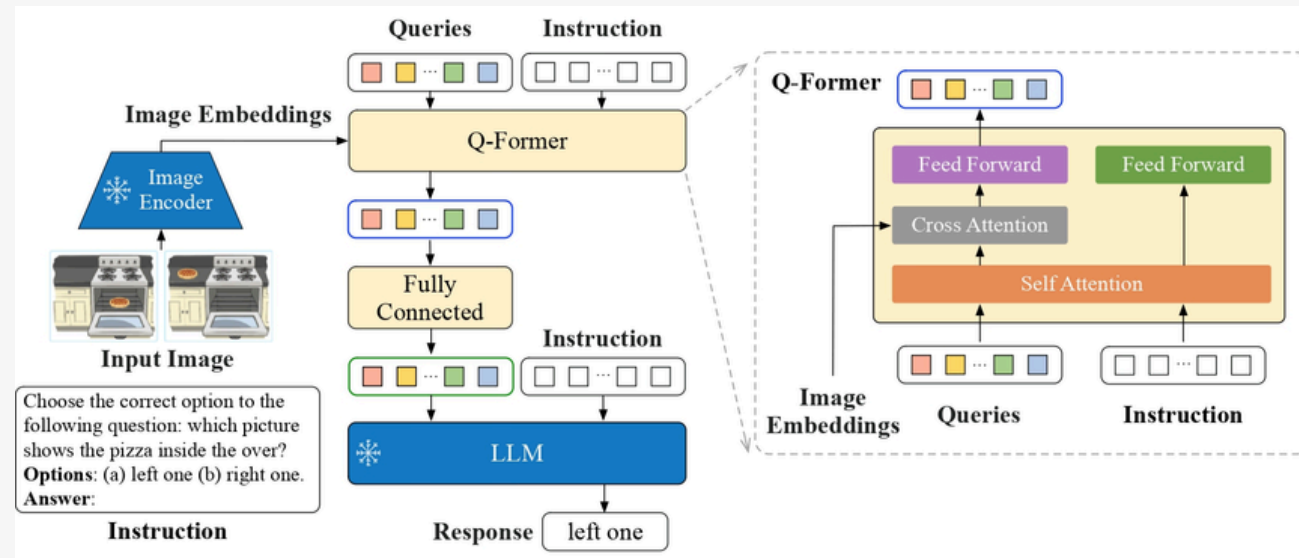
Figure 1. We use a Captioner (Cap) to generate synthetic captions for web images, and a Filter (Filt) to remove noisy captions.

실험과 분석을 통해 다음 2가지를 관찰한다.

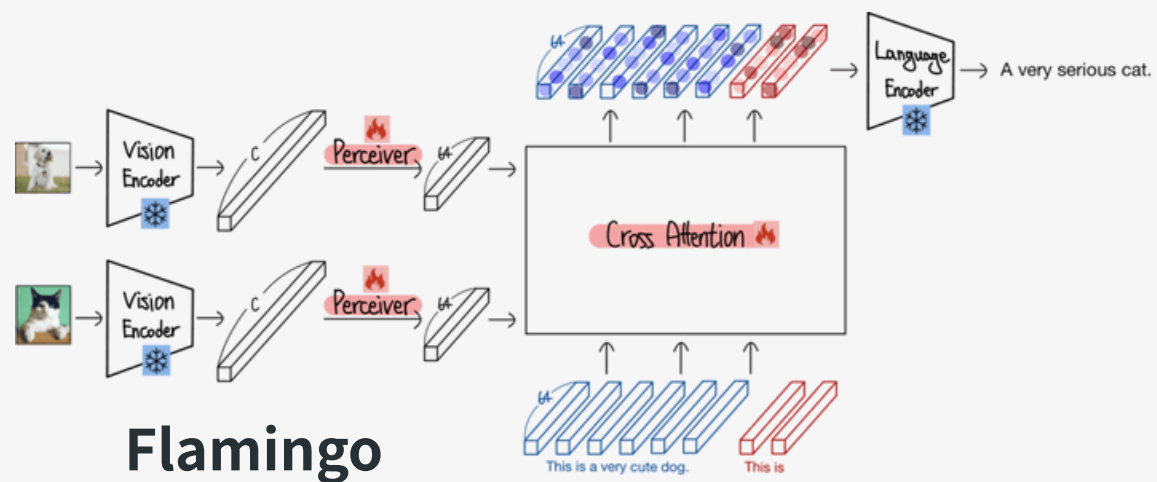
- captioner와 filter가 함께 작동해 captions을 bootstrapping함으로써 다양한 downstream tasks에 상당한 성능 향상을 달성한다. 또 더 다양한 captions이 larger gains을 만든다는 것을 발견했다.
- BLIP은 image-text retrieval, image captioning, visual question answering, visual reasoning, visual dialog을 포함하는 광범위한 vision-language tasks에서 SOTA를 달성한다. 또 2가지 video-language tasks, 즉 text-to-video retrieval과 videoQA에 모델을 직접 transfer했을 때 SOTA zero-shot performance를 달성한다.



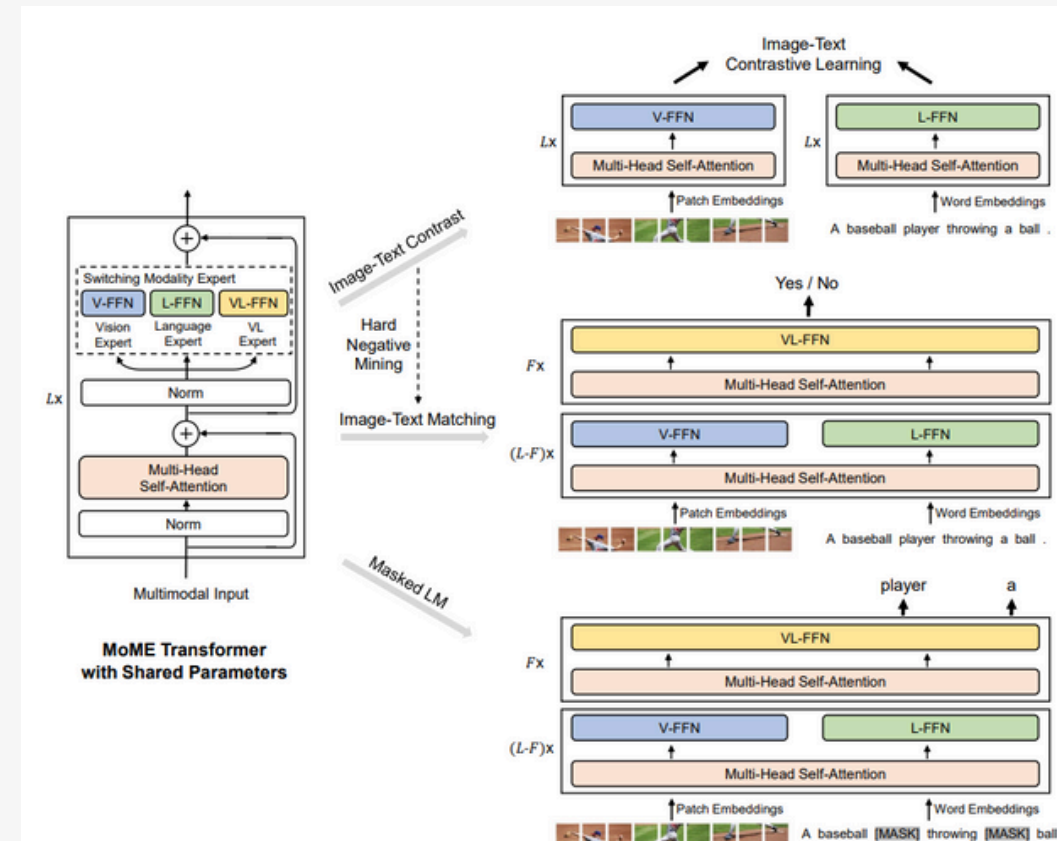
# 01 스터디 활동 리뷰 (논문 분석)



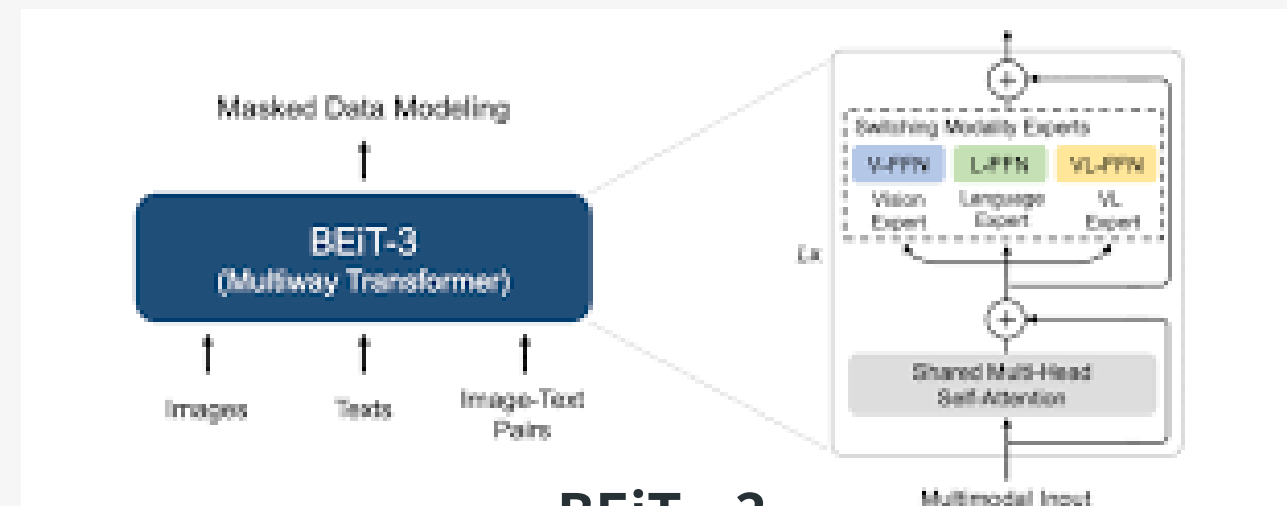
InstructBlip



Flamingo



VLMo



BEiT - 3

## 02 스터디 활동 리뷰 (대회 출전)

### 2025 Samsung Collegiate Programming Challenge : AI 챌린지

채용 | SCPC | 알고리즘 | 비전 | LLM | 생성 AI | 멀티모달 | 정확도

₩ 상금 : 6,000만원

🕒 2025.06.19 ~ 2025.07.28 09:59

+ Google Calendar

👤 1,445명 📅 마감



**대회명: Samsung Collegiate Programming Challenge**

**대회 방법: Summit.CSV를 제출을 통한 점수 채점**



# 02 스터디 활동 리뷰 (대회 출전)

```
import os
import pandas as pd
import cv2
from tqdm import tqdm
import torch
from transformers import Blip2Processor, Blip2ForConditionalGeneration

# 본인에 맞게 분할된 CSV 경로 사용(예: test_part1.csv ~ test_part6.csv)
TEST_CSV = '/content/drive/MyDrive/open/test.csv'
SUBMIT_CSV = '/content/drive/MyDrive/open/submission.csv'
TEST_IMG_DIR = '/content/drive/MyDrive/open/test_input_images'

MODEL_NAME = "Salesforce/blip2-opt-2.7b"
MAX_LEN = 8
BATCH_SIZE = 4
DEVICE = "cuda" if torch.cuda.is_available() else "cpu"

test_df = pd.read_csv(TEST_CSV)
processor = Blip2Processor.from_pretrained(MODEL_NAME)
model = Blip2ForConditionalGeneration.from_pretrained(MODEL_NAME).to(DEVICE)
model.eval()

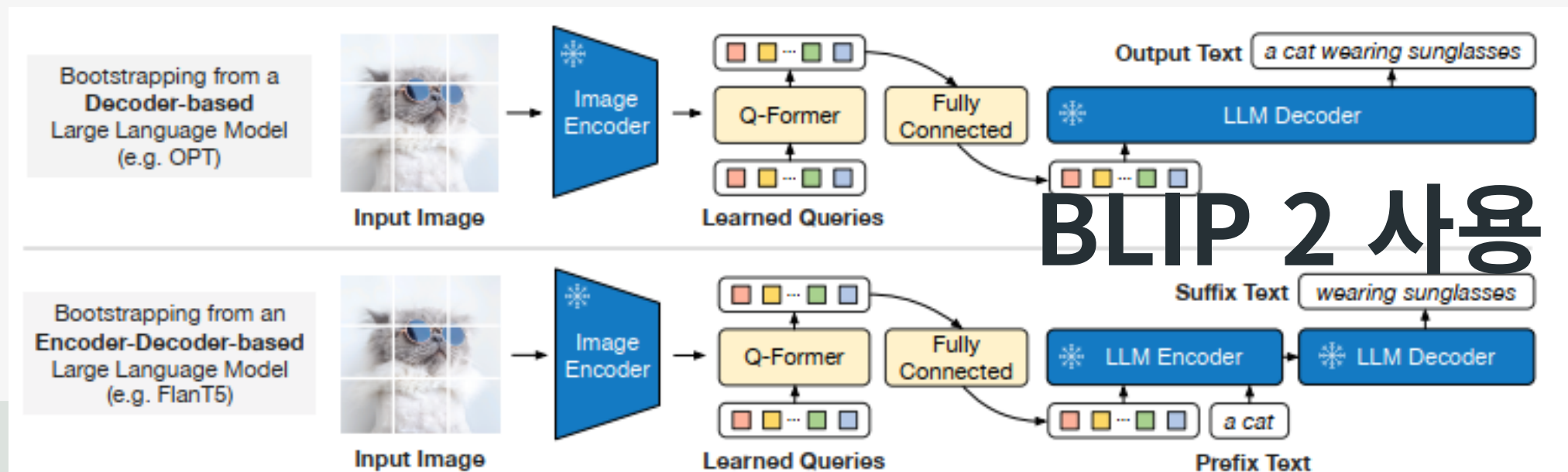
choice_map = {'A', 'B', 'C', 'D'}
id_list, ans_list = [], []

def load_images(paths):
    images = []
    for path in paths:
        img = cv2.imread(path)
        img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
        images.append(img)
    return images

for start in tqdm(range(0, len(test_df), BATCH_SIZE)):
    batch_df = test_df.iloc[start:start+BATCH_SIZE]
    img_paths = [os.path.join(TEST_IMG_DIR, os.path.basename(p)) for p in batch_df['img_path']]
    images = load_images(img_paths)
    prompts = {
        f"Q: {row['Question']} (A){row['A']}, (B){row['B']}, (C){row['C']}, (D){row['D']} Answer:"
        for _, row in batch_df.iterrows()
    }
    inputs = processor(
        images=images,
        text=prompts,
        return_tensors="pt",
        padding="max_length",
        max_length=MAX_LEN,
        truncation=True
    ).to(DEVICE)
    with torch.no_grad():
        out = model.generate(**inputs, max_new_tokens=1)
        preds = processor.tokenizer.batch_decode(out, skip_special_tokens=True)
    for idx, pred in enumerate(preds):
        pred = pred.upper().strip()
        if pred not in choice_map:
            pred = choice_map[0]
        id_list.append(batch_df.iloc[idx]['ID'])
        ans_list.append(pred)

submission = pd.DataFrame({'ID': id_list, 'answer': ans_list})
submission.to_csv(SUBMIT_CSV, index=False)
print(f"제출 완료: {SUBMIT_CSV}")
```


팀	팀 멤버	최종 점수	제출수	등록일
박찬욱	di	0.3198	3	4일 전
dustnehowl		0.91165	9	6일 전
spilab	sp	0.90951	30	3일 전
Carloskim	Ca	0.86683	4	한 달 전
인하대학교 이상혁		0.85659	17	14일 전
고고웅이	고고	0.85659	14	3일 전
Spatz	Sp	0.85531	21	14일 전








## 02 스터디 활동 리뷰 (대회 출전)

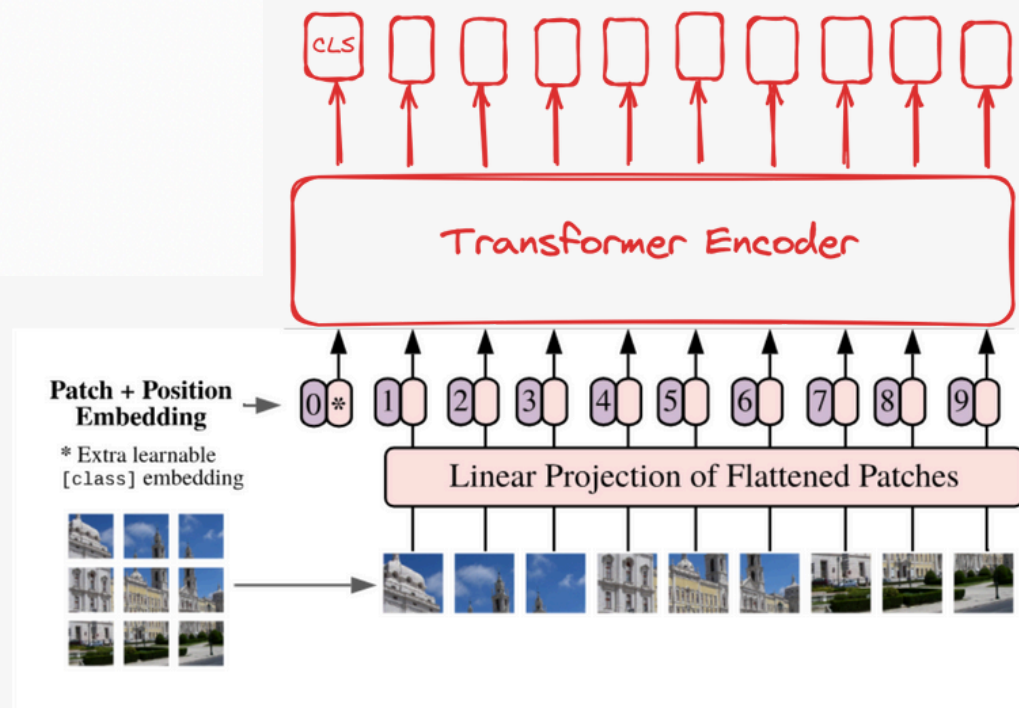
```
# ## Load Pre-trained Model
# 모델 로딩 부분을 google/pix2struct-base로 변경합니다.
# pix2struct-base 모델은 크기가 작아 device_map이나 float16 타입 변환이 필요 없습니다.
processor = Pix2StructProcessor.from_pretrained("google/pix2struct-base")
model = Pix2StructForConditionalGeneration.from_pretrained("google/pix2struct-base")
model.to(device) # 모델을 지정된 장치로 이동합니다.

# ## Model Summary (추가된 부분)
# 불러온 모델의 파라미터 정보를 출력합니다.
total_params = sum(p.numel() for p in model.parameters())
trainable_params = sum(p.numel() for p in model.parameters() if p.requires_grad)
print("✅ Model loaded successfully!")
print(f"Total parameters: {total_params:,}")
print(f"Trainable parameters: {trainable_params:,}")
```

```
Loading widget...  
Loading widget...  
Loading widget...  
Loading widget...  
Loading widget...  
Loading widget...  
Loading widget...  
 Model loaded successfully!  
Total parameters: 282,285,696  
Trainable parameters: 282,285,696
```

김태경		0.26461	3	한 달 전
dustnehowl		0.91165	9	한 달 전
spilab		0.90951	30	한 달 전
Carloskim		0.86683	4	2달 전
인하대학교 이상혁		0.85659	17	한 달 전

# GLT-large-vqa + Pix2Struct 사용



[Submitted on 27 May 2022 (v1), last revised 15 Dec 2022 (this version, v5)]

## GIT: A Generative Image-to-text Transformer for Vision and Language

Jianfeng Wang, Zhengyuan Yang, Xiaowei Hu, Linjie Li, Kevin Lin, Zhe Gan, Zicheng Liu, Ce Liu, Lijuan Wang

In this paper, we design and train a Generative Image-to-text Transformer, GIT, to unify vision-language tasks such as image/video captioning and question answering. While generative models provide a consistent network architecture between pre-training and fine-tuning, existing work typically contains complex structures (uni/multi-modal encoder/decoder) and depends on external modules such as object detectors/taggers and optical character recognition (OCR). In GIT, we simplify the architecture as one image encoder and one text decoder under a single language modeling task. We also scale up the pre-training data and the model size to boost the model performance. Without bells and whistles, our GIT establishes new state of the arts on 12 challenging benchmarks with a large margin. For instance, our model surpasses the human performance for the first time on TextCaps (138.2 vs. 125.5 in CIDEr). Furthermore, we present a new scheme of generation-based image classification and scene text recognition, achieving decent performance on standard benchmarks. Codes are released at [url\[\[this URL\]\(https://github.com/AILab-CVC/GIT\)\]](https://github.com/AILab-CVC/GIT).

Subjects: **Computer Vision and Pattern Recognition (cs.CV)**

Cite as: [arXiv:2205.14100](https://arxiv.org/abs/2205.14100) [cs.CV]

(or [arXiv:2205.14100v5 \[cs.CV\]](#) for this version)

<https://doi.org/10.48550/arXiv.2205.14100> 

# 03 스터디 활동 리뷰 (대회 이후)

## 대회 리뷰

- 대회에서 사용한 코드와 아키텍처 공유
- 한계점을 분석 및 해결 방안 고안
- 해당 해결 방안을 적용하여 어떤 대회에 쓸 수 있을지 조사

## 프로젝트 계획

- 학술제 참가를 목표로 적용가능한 프로젝트 활동 모색



# 04 향후 계획

## 프로젝트 활동

- 다양한 주제를 자유롭게 선택하여 이에 맞는 우리만의 VQA 모델 만들기
- 작년 학술제처럼 팀원끼리 협의해 스스로 흥미로운 주제를 선정
- 선정한 주제에 맞춰 수업에서 배운 내용을 직접 적용해보기
- 결과물을 통해 학습 내용을 실제로 구현하며 학술제를 준비하는 과정

## 학술제 참가

# 05 질문 및 답변

QnA

# Thank you

감사합니다.