LXMERT VQA

Recap

- EMNLP 2019 vision and language model. VQA challenge .
- pre-training expensive pretrained visual BERT LXMERT ( ). LXMERT .
- Pre-training tasks (multi-modality pre-training):
    - masked language modeling
    - masked object prediction (feature regression, label classification)
    - cross-modality matching
    - image question answering
    - Multimodal transformer pre-training
- Dataset used for pre-training
    - COCO caption
    - Visual Genome caption
    - VQA 2.0
    - GQA (Graph Question Answering)
    - Visual Genome QA

Rough Model Architecture

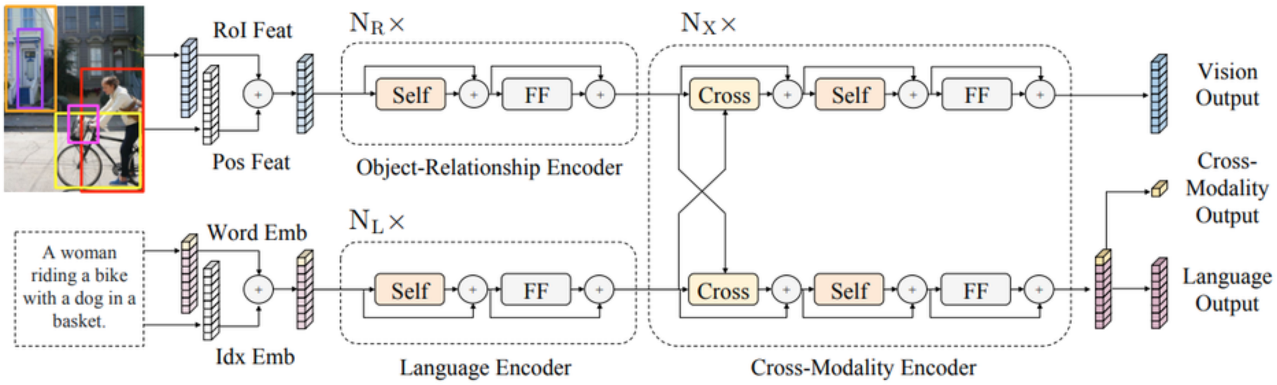
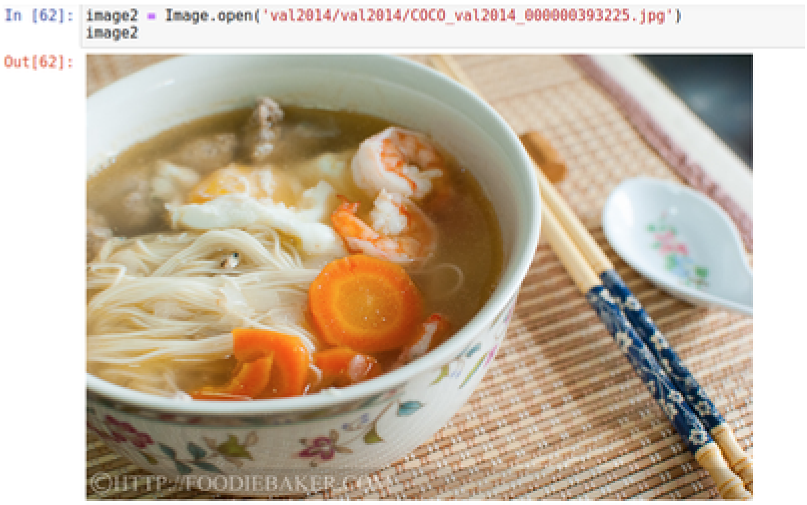


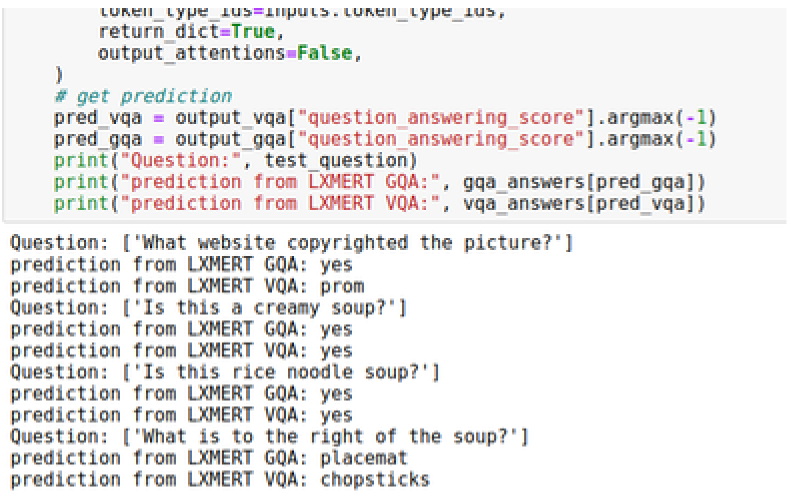
Figure 1: The LXMERT model for learning vision-and-language cross-modality representations. ‘Self’ and ‘Cross’ are abbreviations for self-attention sub-layers and cross-attention sub-layers, respectively. ‘FF’ denotes a feed-forward sub-layer.

Inference

- huggingface lxmert .
- VQA GQA LXMERT huggingface transformers LxmertForQuestionAnswering pretrained weights load .
- Bitbucket .
-
    - tokenizer (input ids, attention mask, token type ids)
    - faster R-CNN roi feature
    - faster R-CNN bounding box (box normalize)
- pre-processing huggingface lxmert tokenizer pre-trained faster R-CNN . huggingface from_pretrained() .
- Inference . VQA challenge 72% (GQA 60%).
- inference .
    -



```
In [62]: image2 = Image.open('val2014/val2014/COCO_val2014_000000393225.jpg')
         image2
Out[62]:
```

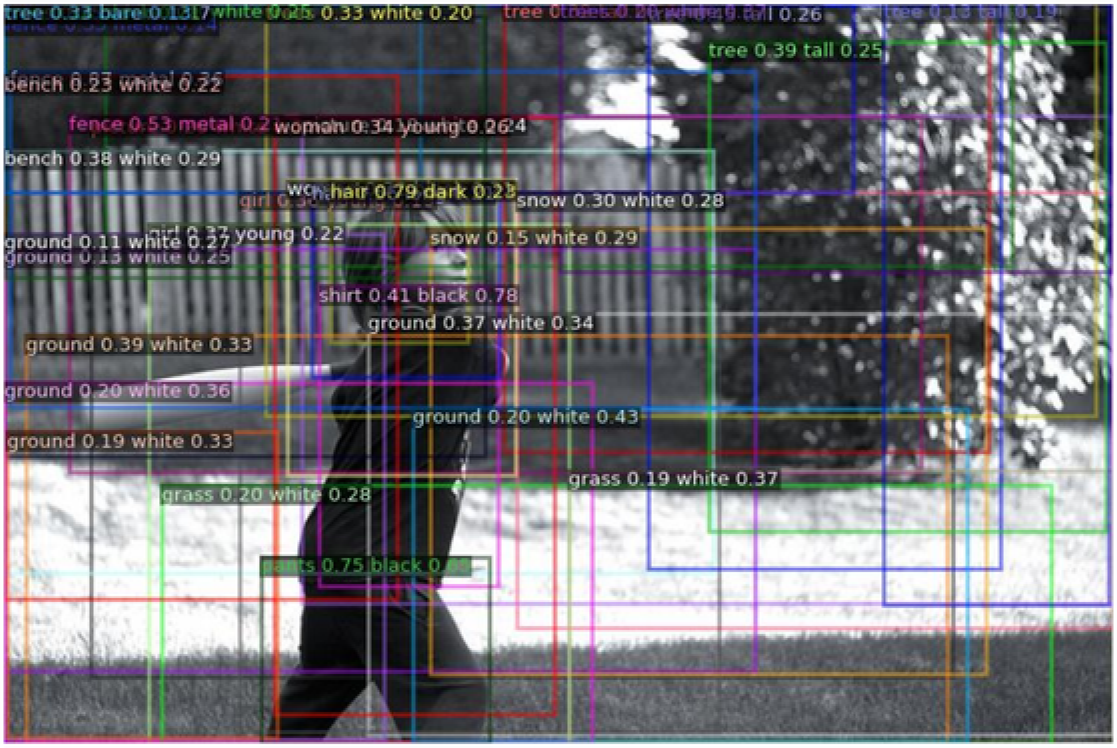
- .
    - 

```
        token_type_ids=inputs.token_type_ids,
        return_dict=True,
        output_attentions=False,
    )
    # get prediction
    pred_vqa = output_vqa["question_answering_score"].argmax(-1)
    pred_gqa = output_gqa["question_answering_score"].argmax(-1)
    print("Question:", test_question)
    print("prediction from LXMERT GQA:", gqa_answers[pred_gqa])
    print("prediction from LXMERT VQA:", vqa_answers[pred_vqa])
```

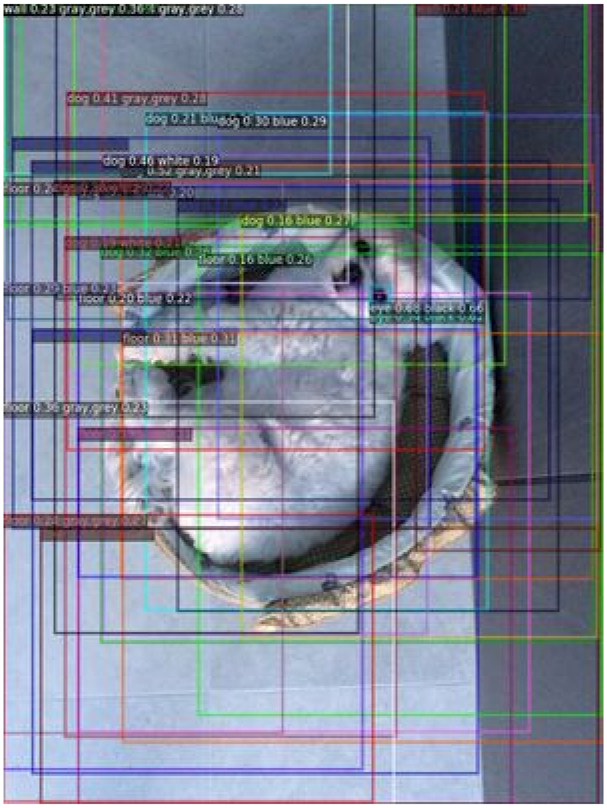
```
Question: ['What website copyrighted the picture?']
prediction from LXMERT GQA: yes
prediction from LXMERT VQA: prom
Question: ['Is this a creamy soup?']
prediction from LXMERT GQA: yes
prediction from LXMERT VQA: yes
Question: ['Is this rice noodle soup?']
prediction from LXMERT GQA: yes
prediction from LXMERT VQA: yes
Question: ['What is to the right of the soup?']
prediction from LXMERT GQA: placemat
prediction from LXMERT VQA: chopsticks
```

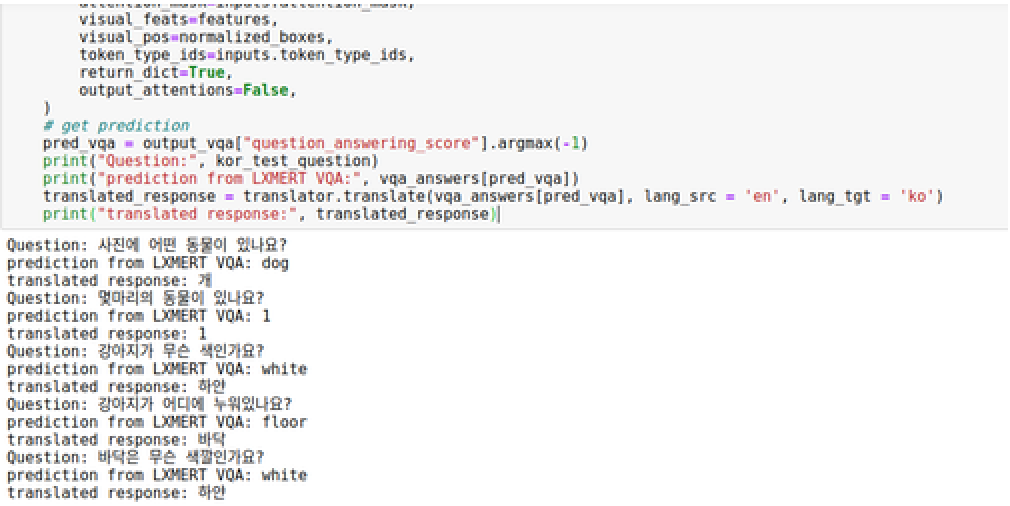
- GQA VQA . VQA task GQA VQA .
- faster R-CNN object detection visualize . Object attribute .
- faster R-CNN directory URL .
    -

```
showarray(frcnn_visualizer._get_buffer())
```



- (VQA v2.0) faster R-CNN feature inference . " ?" ( ) "" . VQA VQA v2.0 8 3192 . , task fine tuning .
  -
  -

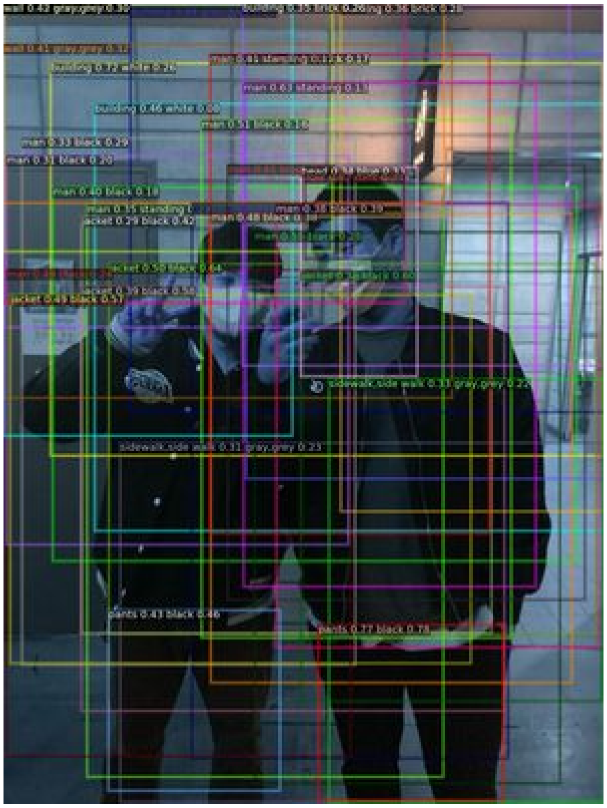


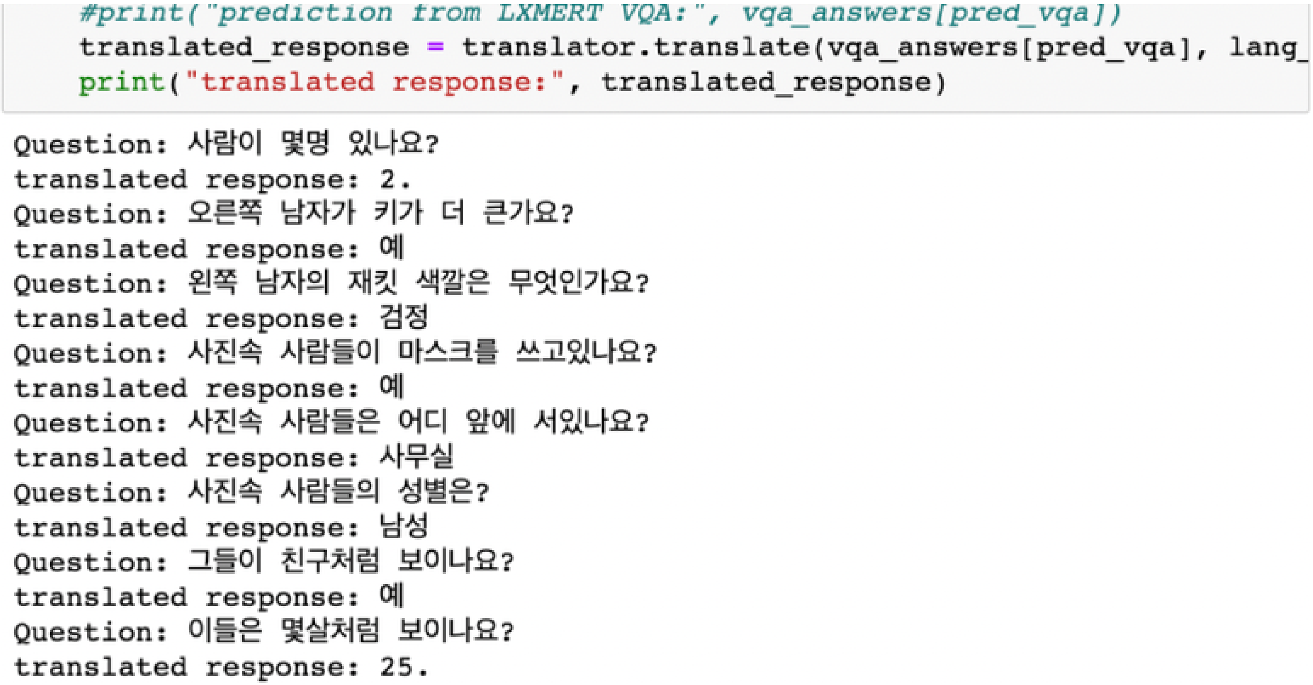


```
        attention_mask=inputs.attention_mask,
        visual_feats=features,
        visual_pos=normalized_boxes,
        token_type_ids=inputs.token_type_ids,
        return_dict=True,
        output_attentions=False,
    )
    # get prediction
    pred_vqa = output_vqa["question_answering_score"].argmax(-1)
    print("Question:", kor_test_question)
    print("prediction from LXMERT VQA:", vqa_answers[pred_vqa])
    translated_response = translator.translate(vqa_answers[pred_vqa], lang_src = 'en', lang_tgt = 'ko')
    print("translated response:", translated_response)
```

```
Question: 사진에 어떤 동물이 있나요?
prediction from LXMERT VQA: dog
translated response: 개
Question: 몇마리의 동물이 있나요?
prediction from LXMERT VQA: 1
translated response: 1
Question: 강아지가 무슨 색인가요?
prediction from LXMERT VQA: white
translated response: 하얀
Question: 강아지가 어디에 누워있나요?
prediction from LXMERT VQA: floor
translated response: 바닥
Question: 바닥은 무슨 색깔인가요?
prediction from LXMERT VQA: white
translated response: 하얀
```

- VQA . . . google_trans_new api . . vqa .
    -


wall 0.42 gray,grey 0.30
building 0.35 brick 0.26
building 0.36 brick 0.28
wall 0.41 gray,grey 0.32
building 0.72 white 0.26
man 0.41 standing 0.12
man 0.63 standing 0.13
building 0.46 white 0.00
man 0.51 black 0.18
man 0.33 black 0.29
man 0.31 black 0.20
head 0.38 blue 0.13
man 0.40 black 0.19
man 0.35 standing
man 0.38 black 0.39
jacket 0.29 black 0.42
man 0.48 black 0.38
jacket 0.50 black 0.64
jacket 0.39 black 0.56
jacket 0.49 black 0.57
sidewalk,side walk 0.33 gray,grey 0.22
sidewalk,side walk 0.31 gray,grey 0.23
pants 0.43 black 0.46
pants 0.77 black 0.78



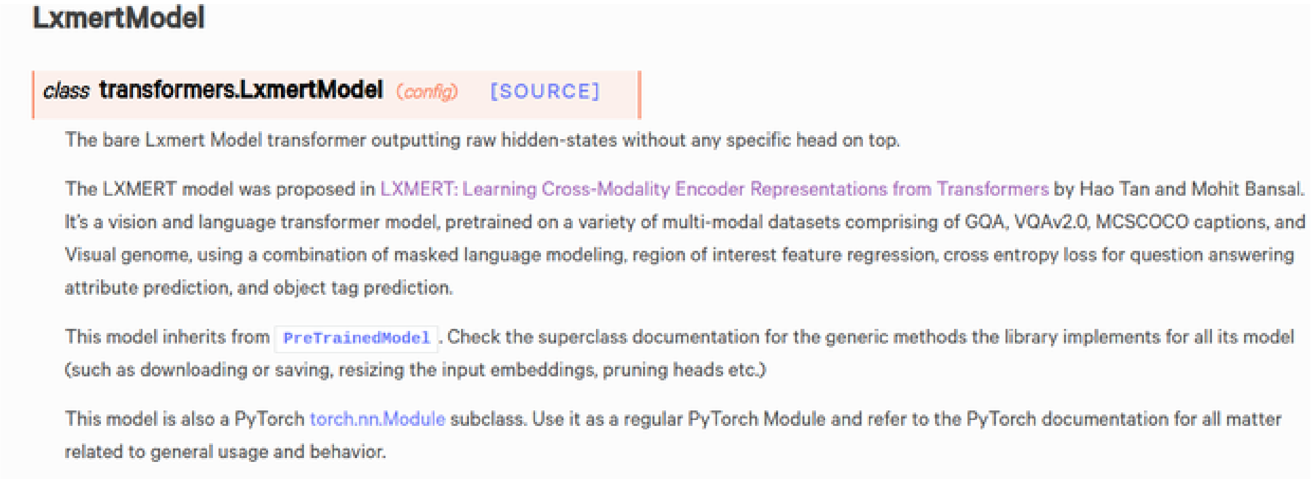
```
    #print("prediction from LXMERT VQA:", vqa_answers[pred_vqa])
    translated_response = translator.translate(vqa_answers[pred_vqa], lang_
    print("translated response:", translated_response)
```

```
Question: 사람이 몇명 있나요?
translated response: 2.
Question: 오른쪽 남자가 키가 더 큰가요?
translated response: 예
Question: 왼쪽 남자의 재킷 색깔은 무엇인가요?
translated response: 검정
Question: 사진속 사람들이 마스크를 쓰고있나요?
translated response: 예
Question: 사진속 사람들은 어디 앞에 서있나요?
translated response: 사무실
Question: 사진속 사람들의 성별은?
translated response: 남성
Question: 그들이 친구처럼 보이나요?
translated response: 예
Question: 이들은 몇살처럼 보이나요?
translated response: 25.
```

-

Fine-Tuning

- VQA huggingface LxmertModel fine-tuning . LxmertModel hidden state output head (output layer) .

  ## LxmertModel

  class **transformers.LxmertModel** (*config*) [SOURCE]

  The bare Lxmert Model transformer outputting raw hidden-states without any specific head on top.

  The LXMERT model was proposed in LXMERT: Learning Cross-Modality Encoder Representations from Transformers by Hao Tan and Mohit Bansal. It's a vision and language transformer model, pretrained on a variety of multi-modal datasets comprising of GQA, VQAv2.0, MCSCOCO captions, and Visual genome, using a combination of masked language modeling, region of interest feature regression, cross entropy loss for question answering attribute prediction, and object tag prediction.

  This model inherits from `PreTrainedModel`. Check the superclass documentation for the generic methods the library implements for all its model (such as downloading or saving, resizing the input embeddings, pruning heads etc.)

  This model is also a PyTorch torch.nn.Module subclass. Use it as a regular PyTorch Module and refer to the PyTorch documentation for all matter related to general usage and behavior.

  -
- Tensorflow 2.0 pytorch pytorch pytorch .
- fine tuning bounding box normalize . . Inference (2-3?) inference .
- Fine-tuning bitbucket : https://pms.maum.ai/bitbucket/users/minsuk_mindslab.ai/repos/lxmert_vqa/browse/lxmert_fine_tuning_example_notebook.ipynb
  - VQA v2.0 fine-tuning . LxmertModel classification head softmax multiclass classification .
  - faster R-CNN image feature . VQA v2.0 10% feature 3 .
  - Faster R-CNN object detection .