Image-Text representation study (based on CLIP and prefix-tuning)

연구 내용 요약

2023.03 김다혜

Image-Text representation study (based on CLIP and prefix-tuning)

연구 동기

- 최근 뛰어난 성능을 보이는 거대 모델 (ChatGPT, DALL-E2 ..)
  하지만, 데이터와 자원에 의존적이라는 한계점이 존재한다.

- 이미지 캡셔닝 모델 -> 대부분 supervised learning으로 많은 labeled data 필요

- 멀티모달 분야에서, **적은 데이터와 한정된 자원으로도 좋은 성능을 낼 수 있는 모델**에 대한 관심
  => ClipCap을 이미지캡셔닝 연구 모델로 삼아 contribution을 찾아보고자 함

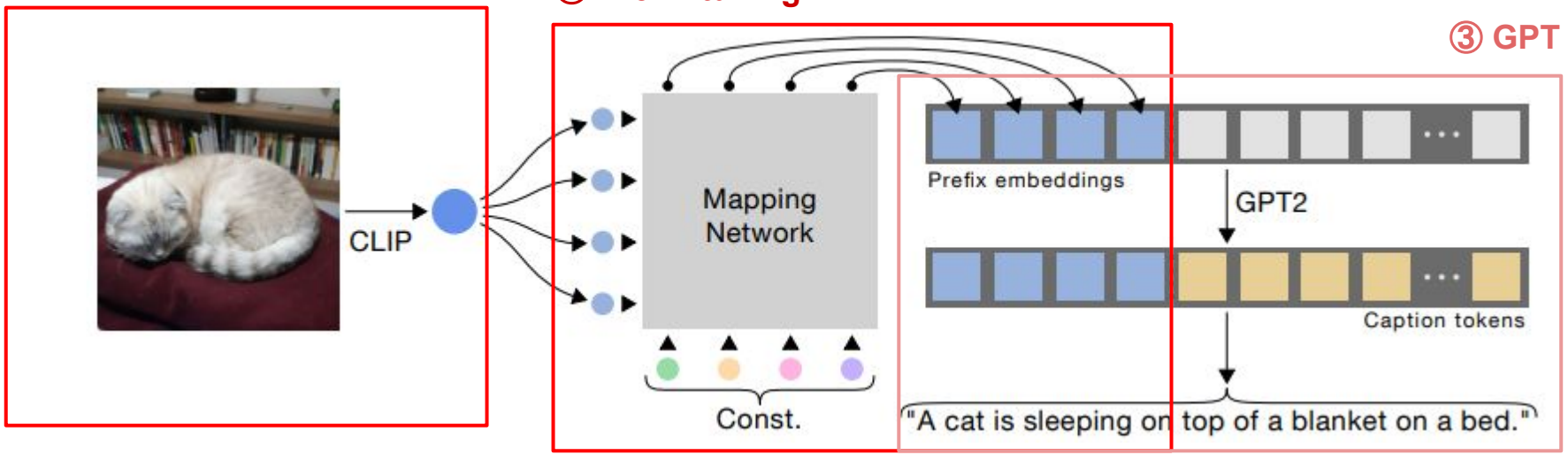
➔ **ClipCap이 적은 파라미터 학습과 간단한 architecture로 좋은 성능을 낼 수 있었던 이유?**

Image-Text representation study (based on CLIP and prefix-tuning)

① CLIP

② Prefix-tuning

③ GPT 2



1) Image embeddings을 얻기 위한 CLIP 인코더,

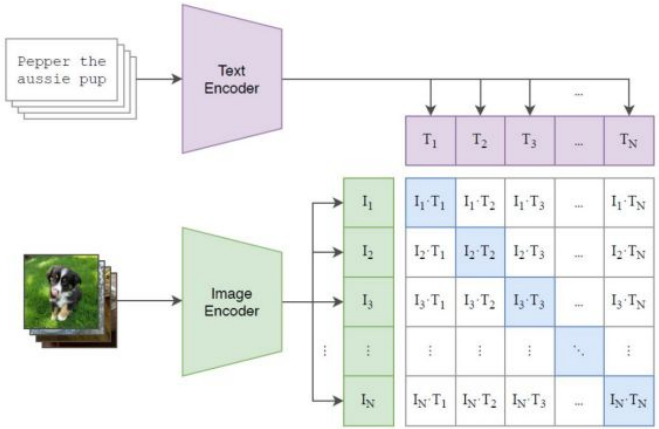
2) prefix tuning을 위한 MLP layer,

3) Text generation을 위한 GPT2가 결합된 형태의 모델

➔ **prefix-tuning** 활용하여 언어모델이 추가 fine tuning 없이도 자연스러운 caption 생성이 가능해짐

Image-Text representation study (based on CLIP and prefix-tuning)

❖ CLIP : Web-based image-text pair 기반으로 visual representation을 사전 학습하는 방법론



- 특징1 : Image-Text pair를 입력으로 사용
  - 일반적인 분류 모델은 이미지의 의미론적 정보를 학습하지 못하는 반면,
  - CLIP은 이미지와 언어에 대한 representation을 함께 학습하여 일반화된 특징 학습이 가능하다. => Visual representation + Sementic information

EX) 일반적인 softmax를 사용한 분류 모델 vs CLIP

기존 분류모델은 이미지의 레이블값을 숫자로 치환해서 softmax 확률값이 가장 큰 레이블을 선택하는데,

이때 레이블 0,1,2,..끼리는 아무런 의미적 관계가 없다.

반면, CLIP은 대조학습을 할때 이미지 인코더와 텍스트 인코더를 모두 사용하기 때문에

레이블에 해당하는 캡션이 트랜스포머 인코더를 통해 임베딩이 되면, 각 텍스트끼리의 의미적 관계를 포함한 임베딩 벡터가 생성된다.

따라서 **CLIP은 레이블 끼리의 유사도 정보까지 미리 pretrain이 되어 더욱 일반화된 표현학습이 가능해짐**!

Image-Text representation study (based on CLIP and prefix-tuning)

- 특징2 : Contrastive Learning으로 효율적인 Pre-train이 가능하다.

- Contrastive Learning = '데이터 내 positive & negative samples 간의 관계 학습'
- 샘플들 각각 임베딩하여 pos(+)샘플 간 유사도는 크게, neg(-) 샘플 간 유사도는 작게 학습한다.
- Contrastive Learning을 통한 유사도 학습은 Zero-shot prediction 에서도 우수한 성능을 보인다.

❖ Prefix-tuning : 언어 모델의 prompt 개념에서 시작한 아이디어

- Prefix : Mapping Network 를 거쳐서 나온 embedding vector
- 언어생성모델인 gpt2가 캡션을 생성하기 전에, 미리 prompt를 제공하기 위해서

  CLIP에서 얻은 이미지 임베딩을 mlp를 거쳐서 prefix embedding 형태로 바꾸어주는 역할을 한다.
- 좋은 성능을 낼 수 있는 prefix값을 뽑기 위해서 mlp의 파라미터값을 최적화하는 parameter tuning 작업
- GPT2의 transformer layer마다(decoder layer마다) prefix를 추가하는 방식이다.

Image-Text representation study (based on CLIP and prefix-tuning)

연구 내용 : 'Prefix 길이에 따른 이미지 캡셔닝 성능 분석'

Clipcap모델이 prefix tuning을 시도한 이유는,

prefix를 생성하기 위한 mlp만 최적화시키고, **clip과 gpt2는 추가 Fine-tuning을 하지 않고도** 전체 모델이 좋은 성능을 보였기때문이다.

논문에서는 prefix의 길이가 증가할 수록 많은 양의 정보가 담겨있다고 함

➔ But, 저자들이 참고했다고 한 prefix-tuning 논문의 실험결과를 보면, prefix길이가 증가한다고 무조건 좋은 성능을 내는 것은 아님

	E2E				
	BLEU	NIST	MET	ROUGE	CIDEr
PREFIX	69.7	8.81	46.1	71.4	2.49
Embedding-only: EMB-{PrefixLength}					
EMB-1	48.1	3.33	32.1	60.2	1.10
EMB-10	62.2	6.70	38.6	66.4	1.75
EMB-20	61.9	7.11	39.3	65.6	1.85
Infix-tuning: INFIX-{PrefixLength}					
INFIX-1	67.9	8.63	45.8	69.4	2.42
INFIX-10	67.2	8.48	45.8	69.9	2.40
INFIX-20	66.7	8.47	45.8	70.0	2.42

Prefix-Tuning: Optimizing Continuous Prompts for Generation

- prefix 길이 증가 –> 무조건 좋은 성능을 가져오는가?

➔ **ClipCap 에서 prefix 길이를 조정하여 성능 비교, 결론 도출**

Image-Text representation study (based on CLIP and prefix-tuning)

❖ Experiment setting

기본 실험 세팅 및 데이터는 ClipCap 논문과 동일하다.

- data = MSCOCO 2014 (train/val)
- epoch = 10
- optimizer = AdamW

❖ 학습 및 추론 과정

- ClipCap에서 사용하는 패키지를 설치하고 CLIP과, GPT2 모델을 불러온다.
- jupyter 가상환경에서 MSCOCO dataset으로 multi layer perceptron optimizing (GPU 환경. 연구실 내 서버 활용)
- prefix length = 1, 10, 20 으로 설정하여 학습 진행
- 각각의 prefix length에 대해, 학습 과정에서 저장된 prefix.pt를 불러와서 추론 진행

CLIP model + GPT2 tokenizer

0초

```
[ ]  #@title CLIP model + GPT2 tokenizer

     device = CUDA(0) if is_gpu else "cpu"
     clip_model, preprocess = clip.load("ViT-B/32", device=device, jit=False)
     tokenizer = GPT2Tokenizer.from_pretrained("gpt2")
```

```
100%|| 338M/338M [00:02<00:00, 136MiB/s]
Downloading: 100% 1.04M/1.04M [00:00<00:00, 4.30MB/s]
Downloading: 100% 456k/456k [00:00<00:00, 4.59MB/s]
```

Load model weights

19초

```
[14] #@title Load model weights

     prefix_length = 20

     model = ClipCaptionModel(prefix_length)

     model.load_state_dict(torch.load('/content/drive/MyDrive/논문/CLIP_prefix_caption/coco_train/coco_prefix-009.pt', map_location=torch.device('cpu')))

     model = model.eval()
     device = CUDA(0) if is_gpu else "cpu"
     model = model.to(device)
```

```
Downloading: 100% 548M/548M [00:06<00:00, 85.3MB/s]
```

❖ Captioning performance (BLEU score)

	BLEU1	BLEU2	BLEU3	BLEU4
ClipCap Paper	-	-	-	32.15
Prefix length = 1	66.13	51.69	36.58	28.42
Prefix length = 10	69.2	53.38	40.74	31.71
Prefix length = 20	68.79	53.4	40.98	30.57

- **prefix length** 별 **BLEU score** 확인 결과, 길이가 증가할 수록 점수가 올라가는 경향성 보인다.
- bleu score1과 4 기준 length=10 일 때 length=20 일 때 보다 미세하게 높음

➔ prefix 길이가 20일 때 무조건 1과 10일때보다 좋은 score를 달성하지는 않는 것을 확인

❖ Samples of qualitative results

length = 10

100%| | 1/1 [00:00<00:00, 5.12it/s]

A bunch of bananas sitting on top of a table.

length = 20

100%| | 1/1 [00:00<00:00, 5.12it/s]

A wooden table topped with lots of ripe and unripe bananas.

A bunch of bananas sitting on top of a table.

A wooden table topped with lots of ripe and unripe bananas.

length = 1	A view of a dining room and chairs.
length = 10	A dining room with a large window and a wooden table.
length = 20	A dining table and chairs in a large room.

length = 1	A dog sitting on top of a brown bench.
length = 10	A white dog sitting on a bench next to a woman.
length = 20	A white and brown dog sitting on top of a wooden bench.

➔ 캡션 인퍼런스를 해봤을 때, prefix 길이가 증가했을때 캡션이 사진을 더 잘 설명하는 것은 아닌 것을 확인

❖ Conclusion

- prefix 길이가 증가하면 캡셔닝 성능이 좋아지는 경향성이 있다.
- 하지만, 길이를 증가하는게 모델 성능을 개선할 수 있는 최적의 방법은 아님

❖ Comparing ClipCap and other SOTA models

(C) **COCO**

Model	B@4 ↑	METEOR ↑	CIDEr ↑	SPICE ↑	#Params (M) ↓	Training Time ↓
BUTD [4]	36.2	27.0	113.5	20.3	52	960h (M40)
VLP [47]	36.5	28.4	117.7	21.3	115	48h (V100)
Oscar [19]	**36.58**	**30.4**	**124.12**	**23.17**	135	74h (V100)
Ours; Transformer	33.53	27.45	113.08	21.05	**43**	**6h** (GTX1080)
Ours; MLP + GPT2 tuning	32.15	27.1	108.35	20.12	156	7h (GTX1080)

- 또한, ClipCap은 **VLP, Oscar**와 같은 거대 사전학습 모델에 비하면 성능이 높지 않다.
- 따라서 ClipCap에서 제시한 prefix-tuning 방법 외에도, decoding 성능을 개선할 수 있는 방법에 대한 연구를 찾아볼 예정

❖ 연구 방향 구체화

For good performance?

1) 적절한 decoding method에 대한 연구

2) Clip과 같이 V-L representation 을 활용한 모델 구조 활용 (good performance at zero-shot)

Research Topic

1) Text generation model & decoding methods

2) zero-shot 에 강한 representation을 활용한 연구 (CLIP)