ClipCap: CLIP Prefix for Image Captioning

논문 리뷰

2023.04.10

INDEX

Introduction

- ClipCap
  - OpenAI 2021.11 발표 논문
  - **Image Captioning task** 해결하기 위한 접근법 중 하나

- **Image captioning ?**

  : 주어진 입력 이미지에 대한 설명 캡션을 예측하여 생성하는 것

  1) 이미지를 통해 개체와 개체간의 관계를 파악할 수 있는 유의미한 특징을 추출
  2) 이미지를 잘 설명할 수 있는 텍스트를 생성

Introduction

“From Show to Tell”

- Image Captioning

ex) “VLP”, “oscar” …

=> '이미지 인코더’ + ‘텍스트 디코더' 구조

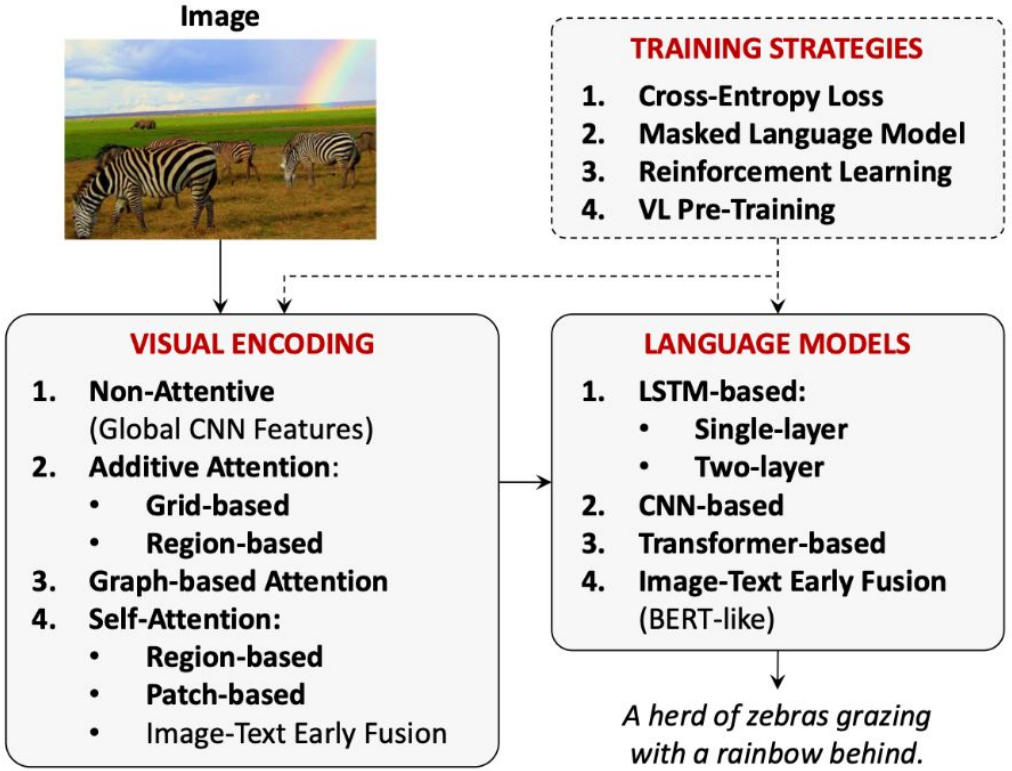


Fig. 1: Overview of the image captioning task and taxonomy of the most relevant approaches.

Introduction

- **이미지 캡셔닝의 주요 과제**

: **"시각적 표현"과 "텍스트 표현"사이의 격차를 해소하는 것**

-> visual-language data간의 데이터의 분포나 특성이 다름

-> 많은 훈련시간, 매개변수, 데이터, 추가 어노테이션 필요…

-> 많은 자원 없이 가벼운 캡션모델로도 좋은 성능 ?

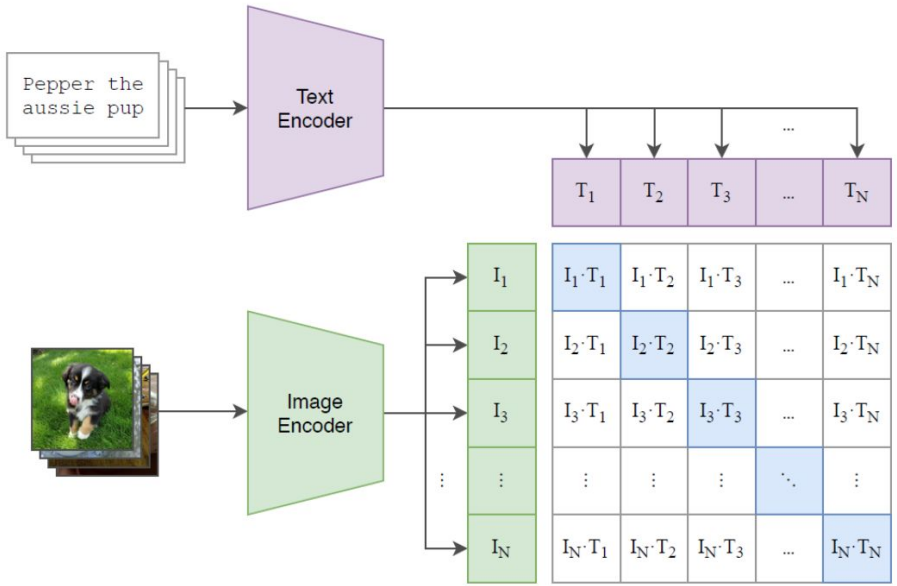
=> CLIPCAP

Method

1) Visual Encoding 과정에서 'CLIP 인코더' 사용

- CLIP : 이미지와 텍스트 프롬프트 모두에 대해 공유표현을 적용

*CLIP의 대조학습 : 이미지와 텍스트가 주어졌을 때, 두 개의 표현이 일치하는지 아닌지 대비하는 식으로 학습



CLIP의 Contrastive pre-training

- image와 text를 하나의 공통된 space로 보낸 다음
- positive pair에서의 코사인유사도는 최대화하고
- negative pair에서의 유사도는 최소화하도록
- CE loss를 사용하여 학습

-> 두 개의 표현이 잘 연관되어 있음

-> 두 표현 사이의 격차 해소

Method

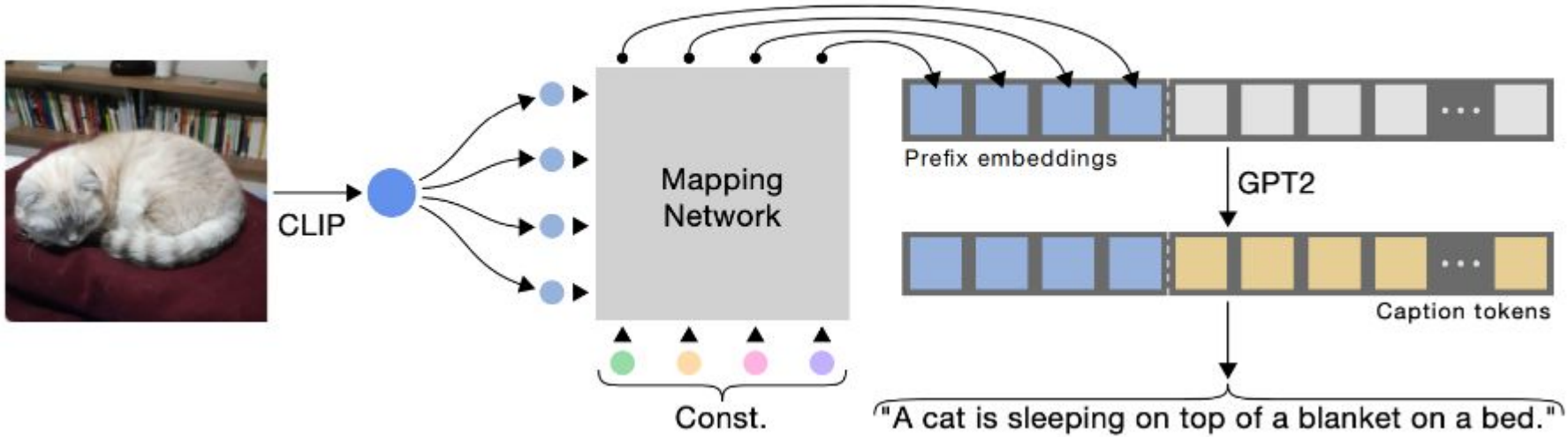
2) 접근방식의 핵심 : prefix

- prefix -> Mapping Network 거쳐서 나온 embedding vector
- prefix를 LM의 인풋으로 넣어줌
- LM이 추가 튜닝 없이 새로운 task에 잘 적용됨

  * inspired by “prefix-tuning”

    *prefix-tuning : LM의 입력 시퀀스에 추가적인 prefix embedding을 결합하여 추가된 embedding 만 학습

Method



1. clip의 이미지 인코더로 feature들을 추출

2. 매핑 네트워크를 거쳐 prefix를 생성

3. 생성된 prefix를 lm에 넣어 prefix로부터 한 단어씩 캡션을 생성

Method

$$p_1^i, \ldots, p_k^i = F(\text{CLIP}(x^i)).$$

-. x : Image embedding

-. k(const) : prefix length

-. p : prefix embedding

- prefix : CLIP과 Mapping Network의 활성화함수를 거쳐 나온 embedding vector
- prefix length (k) = 10

Method

$$\mathcal{L}_X = -\sum_{i=1}^{N}\sum_{j=1}^{\ell}\log p_\theta(c_j^i \mid p_1^i, \ldots, p_k^i, c_1^i, \ldots, c_{j-1}^i).$$

-. c^i : i번째 caption

-. θ : Model's trainable parameters

- 언어모델의 손실함수
- 다음 단어의 확률을 최대화할 수 있는 파라미터값 찾아서 업데이트
- gpt2를 직접 fine tuning하지 않고, gpt2의 입력으로 들어가는 prefix를 학습시키기 위함

Method

학습 과정

1. **CLIP의 visual encoder**로 visual information 추출
2. **Mapping Network**를 통과시켜 embedding vector 추출
3. embedding vector가 **prefix**로 작용하여 caption의 앞쪽에 붙여진다.
4. **GPT-2**를 통해 caption을 생성하고, cross entropy loss를 구한다.
5. back-propagation으로 **Mapping Network를 optimizing**

Experiments

- 최신 SOTA 모델과 비교 (CLIPCAP vs OSCAR, VLP, BUTD)

	Image	*Text*
ClipCap (ours)	CLIP	GPT2
BUTD	Object Detection Network	LSTM
Oscar	Object Detection Network	≈ BERT
VLP	Object Detection Network	≈ BERT

- 데이터셋 : conceptual caption / nocaps datasets / COCO-caption
- 평가 metrics : BLEU, METEOR, CIDEr, SPICE, ROUGE ..

Experiments

Results

(A) **Conceptual Captions**

Model	ROUGE-L ↑	CIDEr ↑	SPICE ↑	#Params (M) ↓	Training Time ↓
VLP	24.35	77.57	16.59	115	1200h (V100)
Ours; MLP + GPT2 tuning	**26.71**	**87.26**	**18.5**	156	80h (GTX1080)
Ours; Transformer	25.12	71.82	16.07	**43**	**72h** (GTX1080)

(B) **nocaps**

Model	in-domain		near-domain		out-of-domain		Overall		Params↓	Time↓
	CIDEr↑	SPICE ↑	CIDEr	SPICE	CIDEr	SPICE	CIDEr	SPICE		
BUTD [4]	74.3	11.5	56.9	10.3	30.1	8.1	54.3	10.1	52	960h
Oscar [19]	79.6	**12.3**	66.1	**11.5**	45.3	**9.7**	63.8	**11.2**	135	74h
Ours; MLP + GPT2 tuning	79.73	12.2	**67.69**	11.26	**49.35**	**9.7**	65.7	11.1	156	7h
Ours; Transformer	**84.85**	12.14	66.82	10.92	49.14	9.57	**65.83**	10.86	**43**	**6h**

(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)

Experiments

Ablation Study

- LM fine-tuning
  - conceptual captions -> fine-tuning 성능이 더 잘 나옴
  - nocaps -> 비슷

- Mapping Network

(D) Ablation						
Ours; Transformer + GPT2 tuning	32.22	27.79	109.83	20.63	167	7h (GTX1080)
Ours; MLP	27.39	24.4	92.38	18.04	32	6h (GTX1080)

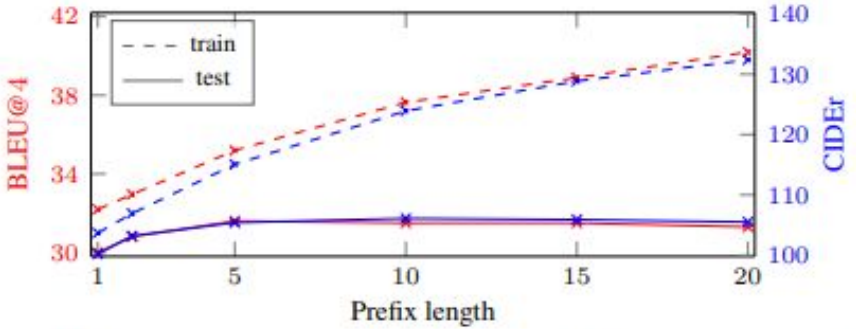
  - Mapping Network 있을 때 보다 성능 떨어짐

Experiments

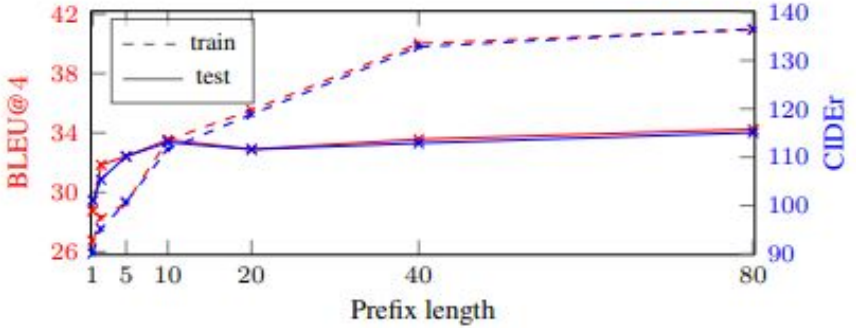
Ablation Study

- **Prefix length**

- LM이 트랜스포머 구조 -> prefix 길이 조정 가능
- prefix length가 길어질수록 많은 양의 정보
  => 성능 증가



(a) MLP mapping network with fine-tuning of the language model



(b) Transformer mapping network with frozen language model.

Conclusion

한계

- CLIP 모델의 한계점 상속 (e.g., 자전거 인식 잘 못함 ?)
- 일부 데이터셋에서 거대 모델보다 성능이 우세하지 못함

의의

- 추가적인 객체 태그 없이 훈련 가능 (oscar는 필요)
- nocaps, conceptual이 coco보다 다양한 시각적 개념을 모델링함

=>