ClipCap

CLIP Prefix for Image Captioning

Luis Eduardo Robles Jiménez

Abstract

A politician receives a gift from politician.

A collage of different colored ties on a white background.

Silhouette of a woman practicing yoga on the beach at sunset.

Aerial view of a road in autumn.

Abstract

Image captioning is a fundamental task in vision-language understanding, where the model predicts a textual informative caption to a given input image. In this paper, we present a simple approach to address this task. We use CLIP encoding as a prefix to the caption, by employing a simple mapping network, and then fine-tunes a language model to generate the image captions. The recently proposed CLIP model contains rich semantic features which were trained with textual context, making it best for vision-language perception. Our key idea is that together with a pre-trained language model (GPT2), we obtain a wide understanding of both visual and textual data. Hence, our approach only requires rather quick training to produce a competent captioning model. Without additional annotations or pre-training, it efficiently generates meaningful captions for large-scale and diverse datasets. Surprisingly, our method works well even when only the mapping network is trained, while both CLIP and the language model remain frozen, allowing a lighter architecture with less trainable parameters. Through quantitative evaluation, we demonstrate our model achieves comparable results to state-of-the-art methods on the challenging Conceptual Captions and nocaps datasets, while it is simpler, faster, and lighter. Our code is available in https://github.com/rmokady/CLIP_prefix_caption.

Introduction

Challenges of the task

“This task poses two main challenges”

- Semantic Understanding
- Large number of ways to describe an image

Challenges of the approaches

“[...] models are resource hungry”

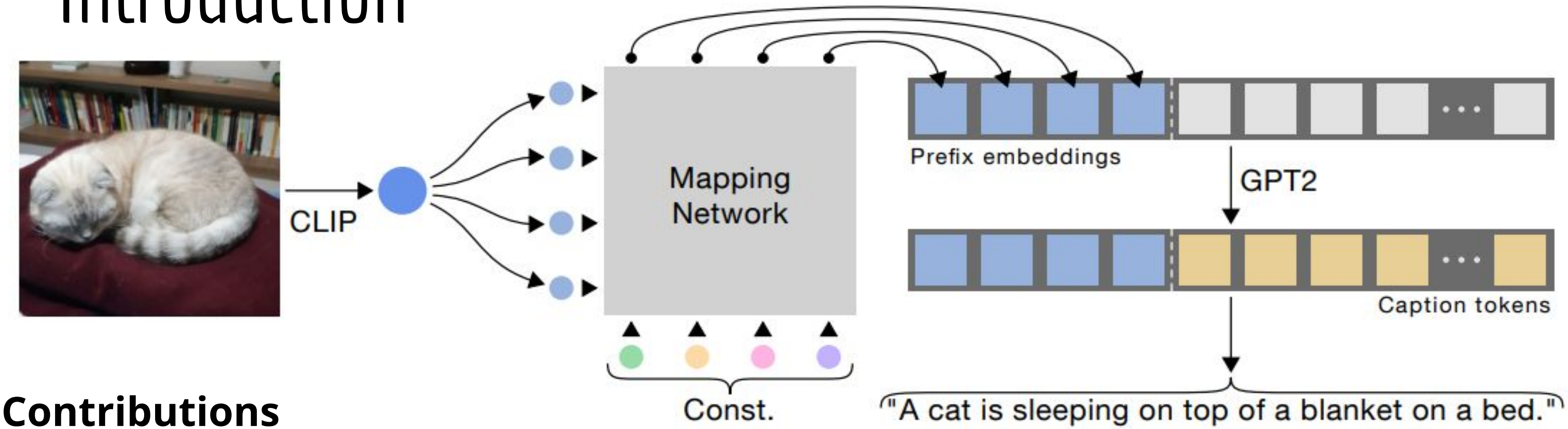
- Extensive training time, large number of trainable parameters and massive datasets are needed.
- A lightweight model is preferable to update the model routinely with the new data.

The idea

In this paper, we leverage powerful vision-language pretrained models to simplify the captioning process.

- *CLIP*
- *GPT-2*

Introduction



Contributions

Overall, our main contributions are as follow:

- A lightweight captioning approach that utilizes pre-trained frozen models for both visual and textual processing.
- Even when the language model is fine-tuned, our approach is simpler and faster to train, while demonstrating comparable results to state-of-the-art over challenging datasets.

Method

3. Method

We start with our problem statement. Given a dataset of paired images and captions $\{x^i, c^i\}_{i=1}^N$, our goal is to learn the generation of a meaningful caption for an unseen input image. We can refer to the captions as a sequence of tokens $c^i = c_1^i, \dots, c_\ell^i$, where we pad the tokens to a maximal length ℓ. Our training objective is then the following:

$$\max_\theta \sum_{i=1}^{N} \log p_\theta(c_1^i, \dots, c_\ell^i | \, x^i), \tag{1}$$

Since the required semantic information is encapsulated in the prefix, we can utilize an autoregressive language model that predicts the next token without considering future tokens. Thus, our objective can be described as:

$$\max_\theta \sum_{i=1}^{N} \sum_{j=1}^{\ell} \log p_\theta(c_j^i | \, x^i, c_1^i, \dots, c_{j-1}^i) \tag{2}$$

Method

3.1. Overview

An illustration of our method is provided in Fig. 2. We use GPT-2 (large) as our language model, and utilize its tokenizer to project the caption to a sequence of embeddings. To extract visual information from an image x^i, we use the visual encoder of a pre-trained CLIP [29] model. Next, we employ a light mapping network, denoted F, to map the CLIP embedding to k embedding vectors:

$$p_1^i, \ldots, p_k^i = F(\mathrm{CLIP}(x^i)). \quad (3)$$

Where each vector p_j^i has the same dimension as a word embedding. We then concatenate the obtained visual embedding to the caption c^i embeddings:

$$Z^i = p_1^i, \ldots, p_k^i, c_1^i, \ldots, c_\ell^i. \quad (4)$$

During training, we feed the language model with the prefix-caption concatenation $\{Z^i\}_{i=1}^N$. Our training objective is predicting the caption tokens conditioned on the prefix in an autoregressive fashion. To this purpose, we train the mapping component F using the simple, yet effective, cross-entropy loss:

$$\mathcal{L}_X = -\sum_{i=1}^{N}\sum_{j=1}^{\ell} \log p_\theta(c_j^i \mid p_1^i, \ldots, p_k^i, c_1^i, \ldots, c_{j-1}^i). \quad (5)$$

We now turn to discuss two variants of our method regarding the additional fine-tuning of the language model and their implications.

Method

Our main challenge during training is to translate between the representations of CLIP and the language model. Even though both models develop a rich and diverse representation of text, their latent spaces are independent, as they were not jointly trained.

3.2 Language Model Fine-Tuning
The style of captioning may not be natural for the pre-trained language model.
Provides flexibility.
More trainable parameters.
Simple architecture.

Even lighter model.
Complex architecture.

Note that fine-tuning CLIP does not benefit resulting quality, but does increase training time and complexity. We hence postulate that the CLIP space already encapsulates the required information, and adapting it towards specific styles does not contribute to flexibility.

Method

3.4. Inference

During inference, we extract the visual prefix of an input image x using the CLIP encoder and the mapping network F. We start generating the caption conditioned on the visual prefix, and predict the next tokens one by one, guided by the language model output. For each token, the language model outputs probabilities for all vocabulary tokens, which are used to determine the next one by employing a greedy approach or beam search.

Results

Evaluation metrics. Similar to Li et al. [19], we validate our results over the COCO dataset using the common metrics BLEU [27], METEOR [10], CIDEr [37] and SPICE [3], and for the nocaps dataset using CIDEr and SPICE. For the Conceptual Captions, we report the ROUGE-L [21], CIDEr, and SPICE, as suggested by the authors [33].

Furthermore, we measure the training time and the number of trainable parameters to validate the applicability of our method. Reducing the training time allows to quickly obtain a new model for new data, create an ensemble of models, and decrease energy consumption. Similar to other works, we report training time in GPU hours, and the GPU model used. The number of trainable parameters is a popular measure to indicate model feasibility.

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			
	CIDEr↑	SPICE ↑	CIDEr	SPICE	CIDEr	SPICE	CIDEr	SPICE	Params↓	Time↓
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

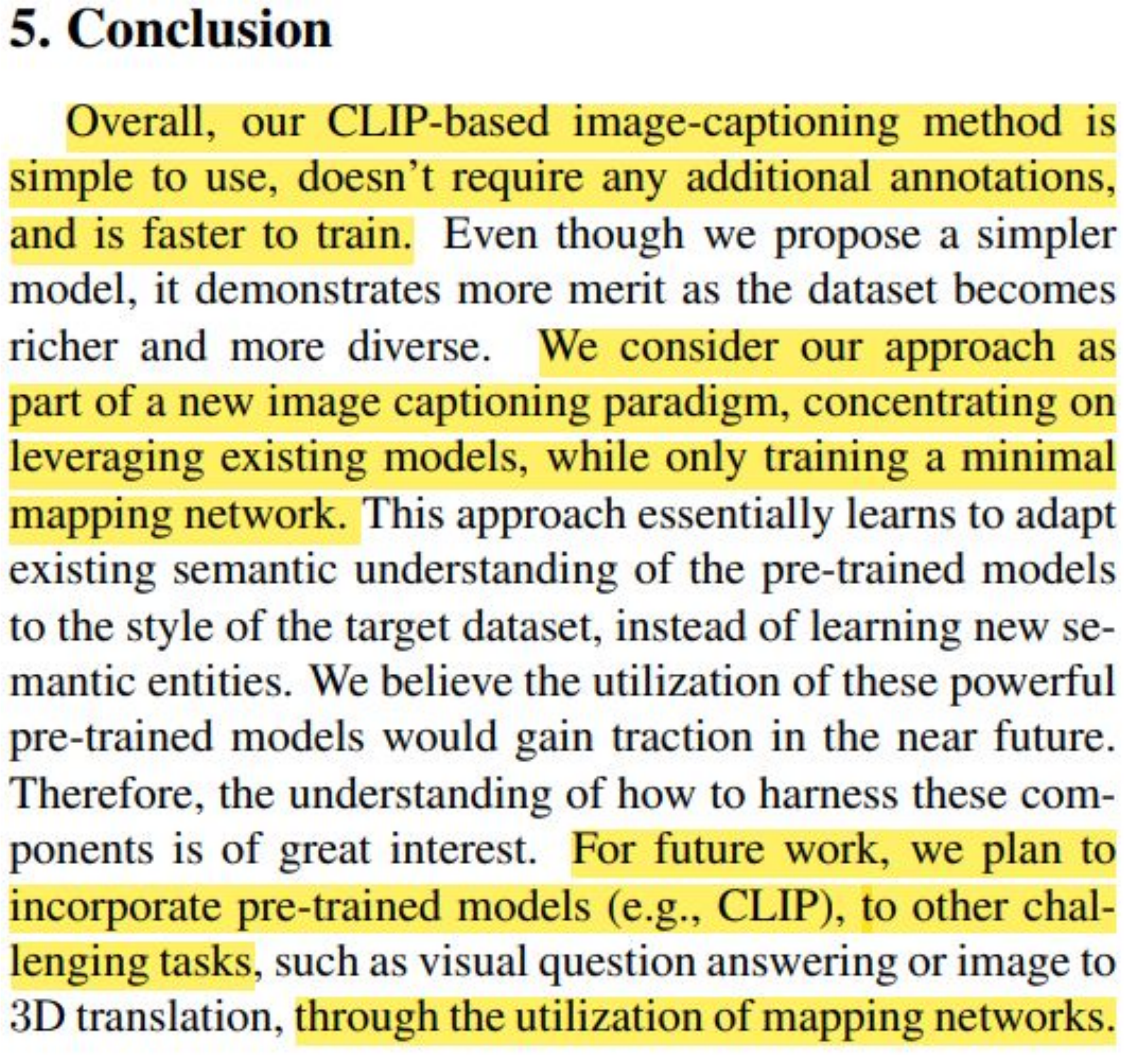
Model	B@4 ↑	METEOR ↑	CIDEr ↑	SPICE ↑	#Params (M) ↓	Training Time ↓
BUTD [4]	36.2	27.0	113.5	20.3	52	960h (M40)
VLP [47]	36.5	28.4	117.7	21.3	115	48h (V100)
Oscar [19]	**36.58**	**30.4**	**124.12**	**23.17**	135	74h (V100)
Ours; Transformer	33.53	27.45	113.08	21.05	**43**	**6h** (GTX1080)
Ours; MLP +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

Results

Ground Truth	A man with a red helmet on a small moped on a dirt road	A young girl inhales with the intent of blowing out a candle.	A man on a bicycle riding next to a train.	a wooden cutting board topped with sliced up food.	A kitchen is shown with a variety of items on the counters.
Oscar	a man riding a motorcycle down a dirt road.	a woman sitting at a table with a plate of food.	a woman riding a bike down a street next to a train.	a woman sitting at a table with a plate of food.	a kitchen with a sink, dishwasher and a window.
Ours; MLP + GPT2 tuning	a man riding a motorcycle on a dirt road.	a woman is eating a piece of cake with a candle.	a man is standing next to a train.	a row of wooden cutting boards with wooden spoons.	a kitchen with a sink, stove, and window.
Ours; Transformer	a man is riding a motorbike on a dirt road.	a young girl sitting at a table with a cup of cake.	a man is standing next to a train.	a wooden table with a bunch of wood tools on it.	a kitchen with a sink and a window.

Figure 3. Uncurated results of the first five images in the COCO test set (Karpathy et al. [17] split).

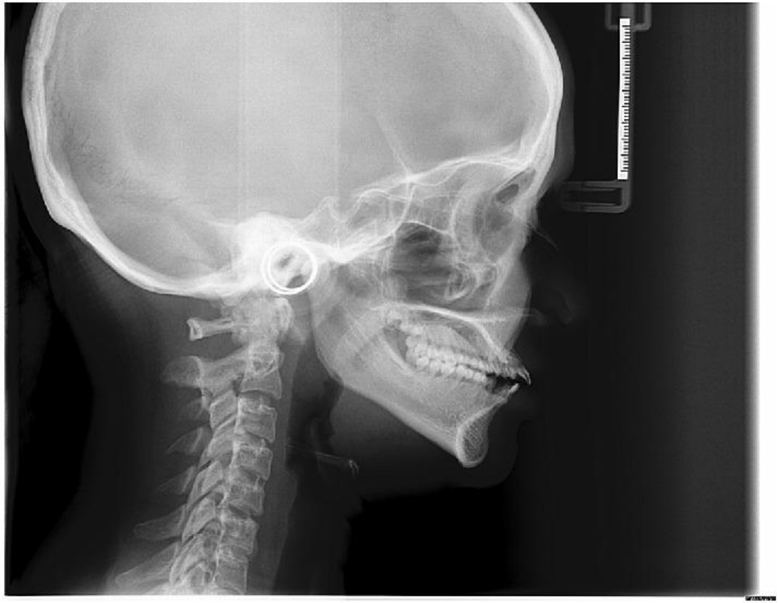
Conclusion



5. Conclusion

Overall, our CLIP-based image-captioning method is simple to use, doesn't require any additional annotations, and is faster to train. Even though we propose a simpler model, it demonstrates more merit as the dataset becomes richer and more diverse. We consider our approach as part of a new image captioning paradigm, concentrating on leveraging existing models, while only training a minimal mapping network. This approach essentially learns to adapt existing semantic understanding of the pre-trained models to the style of the target dataset, instead of learning new semantic entities. We believe the utilization of these powerful pre-trained models would gain traction in the near future. Therefore, the understanding of how to harness these components is of great interest. For future work, we plan to incorporate pre-trained models (e.g., CLIP), to other challenging tasks, such as visual question answering or image to 3D translation, through the utilization of mapping networks.

Own conclusion



Lateral cephalogram of the patient.