Image captioning with text-to-image models

Keith Tyser
Boston University
ktyser@bu.edu

Teona Bagashvili
Boston University
teona@bu.edu

Zihan Li
Boston University
lizihan@bu.edu

Abstract

Various image captioning methods have been proposed using a CNN for feature extraction and an encoder-decoder architecture. However, they fail to capture all of the intricate details in images. Humans are able to easily describe images with great detail but current state-of-the-art models for image captioning fail to generate descriptive enough prompts to reproduce a similar image when fed into a text-to-image model. We propose a novel solution using text-to-image models by formulating a maximum likelihood problem to find the prompt y_{best} that is most likely to recreate an input image z^*. We optimize using gradient descent to learn the continuous embedding of the prompt. When this embedding is fed into Stable Diffusion, we find that the generated images are very similar to the starting input image.

1 Introduction

Automatically generating descriptive captions of images is an important and challenging task. Social media, news, and other online content often includes images. Automatically generating descriptive image captions could be helpful for mining information online, biomedical research, and could also be useful for people with disabilities to interact with the space around them. It's easy for humans to describe the objects and relations in the image, but the majority of the current state of the art image captioning models omit many details.

For example, we used CLIP interrogator [7][5] to generate a starting caption for the first image in Figure 1a. The generated description by CLIP interrogator was "a purple heart sitting on top of a wooden table" Even though this caption is quite descriptive it still omits certain details. For example the material of the object, or the color of the wooden background. Once we gave "a purple heart sitting on top of a wooden table" as a prompt to Stable Diffusion 2 [8], the generated image was not close to the original image, as shown in the second image in Figure 1b. However, if more details are added to the caption such as "a purple metal heart sitting on top of a white wooden table" the new image generated by Stable Diffusion 2 is much closer to the original image as shown in Figure 1c.

Being able to generate more detailed image descriptions can be useful to help recreate any image using text-to-image models. In this work we propose to use the text-to-image model itself to help find the best image caption.

1.1 Related Work

Prior work has explored various image captioning methods. Such as sequentially generating the language description from the image feature vector. They usually extract the features from the image using CNN and then feed this embedding to the encoder-decoder architecture to generate words sequentially [11]. However, a feature vector may not be the best representation of the image, and could lead to captions that aren't as descriptive or accurate.

Majority of the research uses human annotated images for training image captioning models. Hossain et al. proposed using combination of real and synthetic images. The synthetic images were generated

(a) "purple metal heart-shaped object on a white wooden background."

(b) "a purple heart sitting on top of a wooden table."

(c) "a purple **metal** heart sitting on top of a **white** wooden table."

Figure 1: Image (a) was fed into CLIP interrogator to generate an image caption. Image (b) was generated by Stable Diffusion 2 using the caption generated by CLIP interrogator: "a purple heart sitting on top of a wooden table." Image (c) was also generated by Stable Diffusion 2, after a human filled in the prompt with more descriptive words: "a purple metal heart sitting on top of a white wooden table." A more descriptive human-made caption results in an image closer to the original image than the existing state of the art image captioning models. This demonstrates it is easier for humans to describe images than current existing models.

by the text to image model that uses Generative Adversarial Network(GAN). They demonstrated that using synthetic images in addition to real images can be effective for improving the image captioning models [3].

Radford et al. present a method for learning transferable visual models from natural language supervision. The model uses a convolutional neural network (CNN) combined with a recurrent neural network (RNN) to learn visual features from text descriptions. The authors propose a training procedure where the CNN and RNN are trained jointly, and the CNN is used to generate visual features for the RNN to learn from. These features are then used to classify images into categories, and the model is evaluated on several public datasets. The results show that the model is able to transfer knowledge across different domains, and outperforms existing methods. [7]

A novel pre-training approach for unified vision-language understanding and generation is presented by Li et al. [5] The proposed method, called BLIP, exploits the information contained in both images and text to pre-train a vision-language model to better understand and generate image-text pairs. The BLIP model is trained on a large-scale image-text corpus, and is capable of learning a representation of both language and vision that is more robust and generalizable than the representations learned by traditional models. To further enhance the model's generalization ability, BLIP introduces a bootstrapping mechanism that uses the synthetic images generated by the model to further refine the model's language-image pre-training. The authors evaluate the BLIP model on multiple benchmark datasets and show improved performance over existing methods.

Our approach differs from other related work because we do not use a pretrained neural network, but instead solve an optimization problem at inference time.

2 Problem Statement

Using the same notations as in the Stable Diffusion paper [8], the diffusion model represents the probability $p(z|y)$ where y is context data (i.e. text in our application), and z an image in latent space (rather than pixel space). For a fixed y, we can sample from $p(z|y)$, and given a pair $\{z, y\}$, we can approximate the likelihood $p(z|y)$.

Given an image z^*, we want to find the text prompt y_{best} that is the most likely to have generated it. Therefore, we want to solve the Maximum Likelihood problem:

$$\arg\max_y p(z^*|y). \tag{1}$$

This can be solved using gradient descent and will give us the text prompt that is the most likely to have generated the image and this can be done since the diffusion model is differentiable.

3 Methods

Prompt	Log Likelihood
"a red square in the center of the image, black background"	**-31.91**
"a red circle in the center of the image, black background"	-32.54
"a green square in the center of the image, black background"	-35.09

Table 1: To ensure the log likelihood calculations made sense, the likelihood of text-image pairs was calculated. An image of a red square in the middle of the image with a black background for testing which can be seen in Figure 2. We computed p(img | "a green square in the center of the image, black background"), p(img | "a red square in the center of the image, black background"), and p(img | "a red circle in the center of the image, black background"). The likelihood for the correct prompt "a red square in the center of the image, black background" is much lower than the incorrect prompt "a green square in the center of the image, black background", however the likelihood between the first and second prompts are close because there is a high diversity in the produced images and therefore the difference between a square and a circle is tiny. The samples generated will average during gradient descent however which should negate the impact.

We modify Stable Diffusion v1.4 to compute $p(z|y)$. The negative log-likelihood cannot be computed exactly, but the lower bound on the log-likelihood similar to variational auto-encoders [4] can be approximated. The formula for the lower bound is the following:

$$L \geq K$$

$$K = -\sum_{t=2}^{T} \int d\mathbf{x}^{(0)} d\mathbf{x}^{(t)} q\left(\mathbf{x}^{(0)}, \mathbf{x}^{(t)}\right)$$

$$D_{KL}\left(q\left(\mathbf{x}^{(t-1)} \mid \mathbf{x}^{(t)}, \mathbf{x}^{(0)}\right) \| p\left(\mathbf{x}^{(t-1)} \mid \mathbf{x}^{(t)}\right)\right) + H_q\left(\mathbf{X}^{(T)} \mid \mathbf{X}^{(0)}\right) - H_q\left(\mathbf{X}^{(1)} \mid \mathbf{X}^{(0)}\right) - H_p\left(\mathbf{X}^{(T)}\right)$$

The entropy is constant and can be omitted. The lower bound can therefore be computed to a constant factor:

$$\sum_{t=2}^{T} \mathbb{E}_{q(x^{(0)}, x^{(t)})} D_{KL}\left(q\left(\mathbf{x}^{(t-1)} \mid \mathbf{x}^{(t)}, \mathbf{x}^{(0)}\right) || p\left(\mathbf{x}^{(t-1)} \mid \mathbf{x}^{(t)}\right)\right)$$

and can be approximated with Monte Carlo integration:

$$\sum_{t=2}^{T} \sum_{m}^{M} D_{KL}\left(q\left(\mathbf{x}^{(t-1)} \mid \mathbf{x}^{(t)}, \mathbf{x}^{(0)}\right) || p\left(\mathbf{x}^{(t-1)} \mid \mathbf{x}^{(t)}\right)\right) \tag{2}$$

Since $q(x_{t-1}|x_t, x_0)$ and $p(x_{t-1}|x_t)$ are Gaussian distributions, the KL divergence is computed analytically based on the mean and standard deviation of the distributions. The standard deviation of $p(x_{t-1}|x_t)$ is 0, therefore we only use the mean of the distributions, and thus the KL divergence is $||mean_1 - mean_2||_2^2$. The mean and standard deviation of $q(x_{t-1}|x_t, x_0)$ is computed using Bayes Theorem [10]. Our final loss function is the following:

$$\sum_{m} \sum_{t} ||\mu_1 - \mu_2||_2^2$$

We use only one Monte Carlo integration step, similar to how variational auto-encoders [4] and diffusion models [2] are trained. Before optimizing, we verify that the log likelihood calculations are being done correct which can be seen in Table 1. Once the Stable Diffusion code was modified to compute the lower bound of the log likelihood, we use gradient descent and backpropagation to optimize and learn the continuous text embedding most likely to recreate the starting image z^*.

4 Results

An image of a red square in the middle of the image with a gray background was used for experimenting as shown in Figure 2. The starting prompt was initialized to be "a room" so that it is very different from the correct prompt. However, other initialization methods could be experimented with. The starting image from the initialized prompt can be shown in Figure 3.

Figure 2: The image used for testing that we are trying to learn the prompt y_{best} for that will recreate this image when fed into Stable Diffusion. It is an image of a red square in the middle of the image with a gray background.

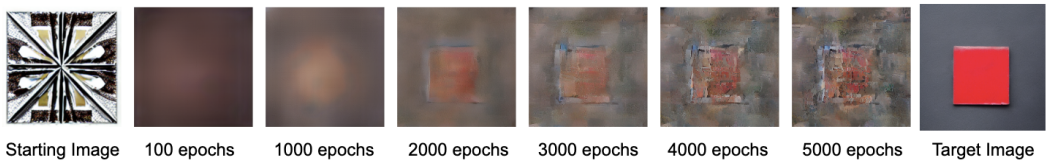

Figure 3: From left to right is the progression of generated images from the learned continuous text embedding during training. The starting image from the prompt "a room" is on the far left, and the target image which is trying to be learned is on the far right

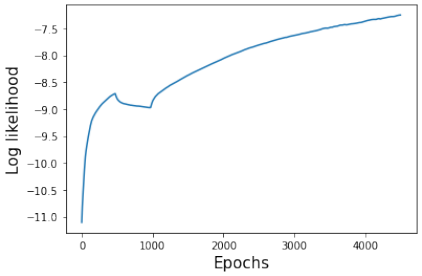

Figure 4: Plot of the log likelihood over 5000 epochs. The log likelihood steadily increases during training. Gradient clipping was used to help prevent exploding gradients which were initially happening during training. This can be seen in Figure 5.

The generated images from the learned continuous embedding after 5000 epochs of training can be seen in Figure 3. It takes approximately 1 hour of training to complete 1000 epochs. After 1000 epochs, the log likelihood is approximately -9, after 2000 epochs, it is approximately -8.25, after 3000 epochs, it is approximately -7.9, after 4000 epochs, it is approximately -7.5, and after 5000 epochs, it is approximately -7.25. The log likelihood consistently increases throughout training except for a slight dip which can be seen around 500 epochs. Over the course of 5000 epochs, we can see a red square is starting to form in the center of the images just like the target image. The red square becomes more defined throughout training, and the background starts to become closer to the color of the target image. It is clear as the log likelihood increases during training, the generated images become closer to the target image.

5 Future Work

5.1 Exploding Gradient

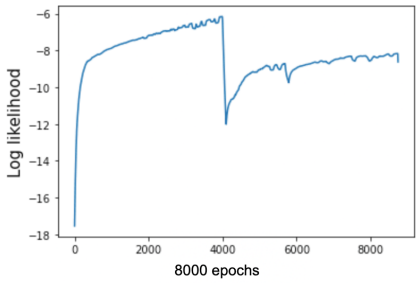

Figure 5: Plot of the log likelihood over 8000 epochs. A sharp collapse can be seen in the log likelihood after 4000 epochs.

One issue we were running into was a collapse of the log likelihood during training which can be seen in Figure 5. This is likely due to the gradient exploding during training. Due to this problem, it was hard to train our model for long periods of time because the model would lose its training progress every few hours from the exploding gradient. We tried to resolve this issue using gradient clipping and had some success which can be seen in Figure 4. With gradient clipping, the model was able to train with less collapses, however, more improvements could be made.

5.2 Converting Continuous Embedding to Tokens

Another area of our research which could be greatly expanded upon is converting the learned continuous embedding to discrete tokens. Because we are only able to update the embedding with gradient descent and not the text, we are not able to see the final prompt that is learned after training. Our model outputs a learned continuous embedding v. A prompt p with embedding v_p can be found such that $d(v, v_p)$ is small. Cosine similarity can be used as the distance function and the function can be optimized using Bayesian Optimization [9]. Doing so would allow the model to output the final learned prompt in a human-readable form, which could facilitate a better understanding of the model's behavior and improve its interpretability.

5.3 Improving Training Efficiency

The main limitation of our methodology is the slow training time. This is because we need to do many gradient steps and because diffusion models are slow in general as they are sequential. In the future, we want to bring our methods to only a few gradient steps, which will reduce the optimization process to seconds. This can be done using meta-learning. A general starting prompt p* can be found such that starting from this prompt, only a few gradient steps will be needed [6] [1]. Reducing the number of gradient steps would significantly improve the efficiency of the methodology, allowing it to be used for larger and more complex image captioning tasks.

6 Conclusion

We make significant progress in image captioning using a novel approach that uses text-to-image models. We formulate a maximum likelihood problem to find the prompt y_{best} that is most likely to recreate an input image z^*. Using gradient descent, we are able to learn the continuous embedding of the prompt. We find that when this learned embedding is fed into Stable Diffusion, generated images are very similar to the target image, as shown in Figure 3. There is a lot of room for future work which includes improving the training efficiency, mitigating the effects of the exploding gradient problem, and converting the learned continuous embedding to text.

References

[1] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pages 1126–1135. PMLR, 2017.

[2] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

[3] Md Zakir Hossain, Ferdous Sohel, Mohd Fairuz Shiratuddin, Hamid Laga, and Mohammed Bennamoun. Text to image synthesis for improved image captioning. *IEEE Access*, 9:64918–64928, 2021.

[4] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[5] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. *arXiv preprint arXiv:2201.12086*, 2022.

[6] Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*, 2018.

[7] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021.

[8] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021.

[9] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical bayesian optimization of machine learning algorithms. *Advances in neural information processing systems*, 25, 2012.

[10] Lilian Weng. What are diffusion models? *lilianweng.github.io*, Jul 2021.

[11] Mingrui Wu, Xuying Zhang, Xiaoshuai Sun, Yiyi Zhou, Chao Chen, Jiaxin Gu, Xing Sun, and Rongrong Ji. Difnet: Boosting visual information flow for image captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18020–18029, 2022.