W266 - Spring 2020

Annabelle Lee

Natalie Wang

GPT-2 Poem Generation from Images

Abstract:

In this project, we created a program that generates poems from images. There are two main parts of our program; first, an object detection step where we determine what is in the given image; and second, a poem generation step where we use a natural language processing model, GPT-2, to generate a poem according to what is in the image. This program could be used to create new poetry or help in the poetry-making process.

Introduction:

Our goal was to generate readable poetry from images. We decided on this topic because poetry doesn't require standard syntax/grammar/vocabulary, and we thought that might be easier to achieve for a model. Also, we thought it could potentially be used to help write poetry in the future; for example, machines don't really get tired so they could be used to generate many samples to select from, then a human could choose the best one or work on them from there. We chose to start from pictures rather than words for the initial input because sometimes when you listen to poetry you can visualize strong images from the poem and we wanted to see if we could do that backwards and generate the poems from the images.

Additionally, computers are good at tasks like calculating and looking for objectively correct answers. However, we want to prove that using natural language processing techniques, it can also perform well in the creative field -- literature.

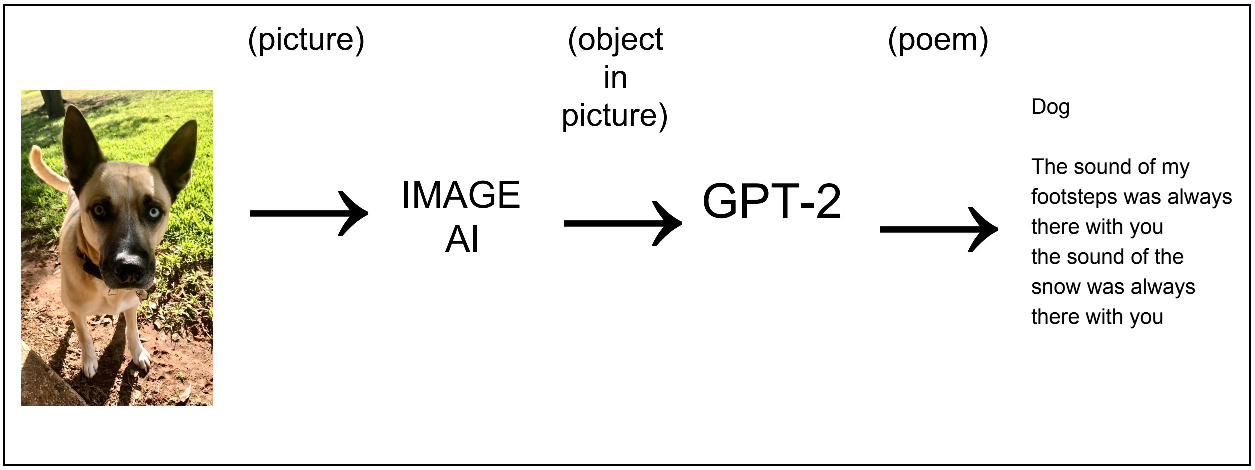
Background:

This step we used an open source library ImageAI to conduct object detection. ImageAI uses RetinaNet to train on image data in JPG format and output objects detected in a given photo. The output contains objects with probability over 0.5. We used this model not only because it achieves the state-of-the-art performances and outperforms the well known two stage detectors such as Faster-R-CNN[1].

The next step is the poetry generation using GTP-2. GPT-2 stands for "Generative Pretrained Transformer 2", it is a transformer-based language model created by OpenAI trained to predict the next word of Internet text. GPT-2 has over 1.5 billion parameters and was trained on a dataset of 8 million web pages [2]. Deep learning and pre-trained models have demonstrated excellent results in natural language processing tasks. In particular, fine-tuning the pre-trained models such as ELMo, OpenAI GPT, GPT-2 and BERT reaches state-of-the-art results [3]. Among the deep learning models above, we chose GPT-2 because it is the newest version of GPT and has outperformed BERT on text generation tasks [4]. In fact, GPT-2 is so powerful that OpenAI only released a smaller version of their original model due to concerns about potential malicious uses of their technology. In this project, we used their smallest GPT-2 model to generate poetry from text inputs.

Methods:

Our approach consisted of an object detection model and a poem generation model. Please see Figure 1 for a diagram of our approach.



Dog

The sound of my
footsteps was always
there with you
the sound of the
snow was always
there with you

FIGURE 1: RESNET + GPT-2 Diagram

For object detection, we imported ImageAI and used it to generate keywords in a given image as inputs to the GPT-2 model. We only used the top predicted objects with probabilities that appeared in the image over 0.8.

The first step for the GPT-2 model was to prepare our dataset. We used data from Project Gutenberg, a public library with over 60,000 free ebooks. It contains about 3 million lines of poetry. Because GPT-2 doesn't have any "memory", we did not want unnecessary formatting or whitespaces to take extra space. Thus, we stripped any leading and trailing whitespaces in the text. Next, we added end of text tokens between each poem, "\n<|endoftext|>\n", so the model could hopefully learn more about poem flow and structures. The last step in data preparation was to convert from a txt file to a npz file for better data compression. All of these steps were done in the json_data.ipynb notebook.

After we had our dataset, we began training the model using the train.py script. First, we worked on fine-tuning the learning rate. During training, our GPT-2 model reports log loss and average loss of cross-entropy after each training step. We trained our model with different learning rates and plotted the average loss after 100 steps. Please see Figure 2 for this graph.

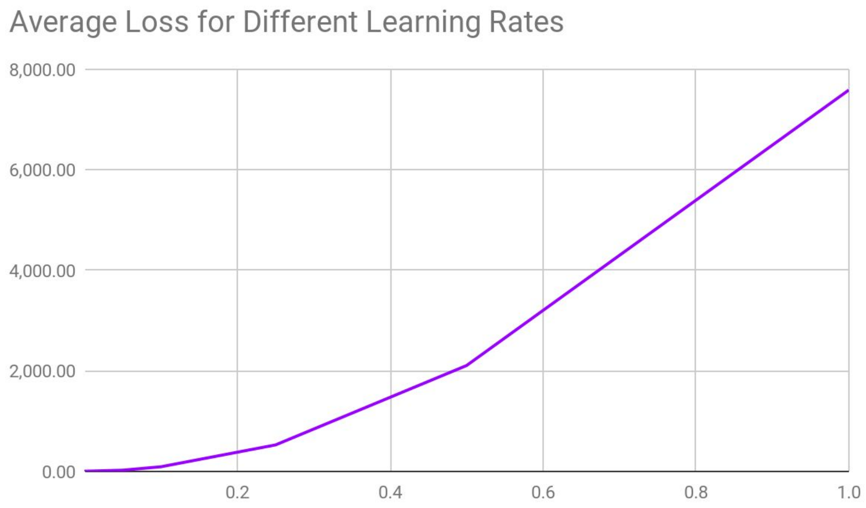


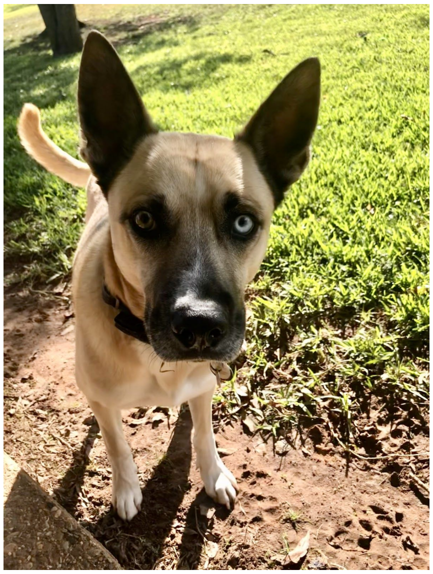
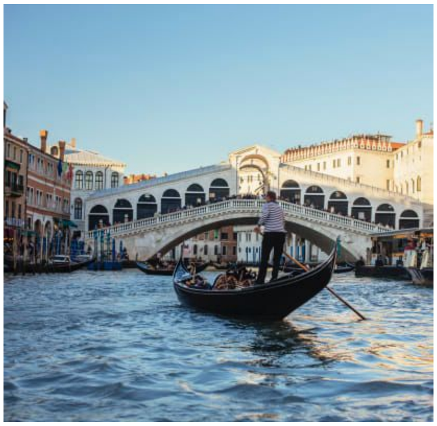
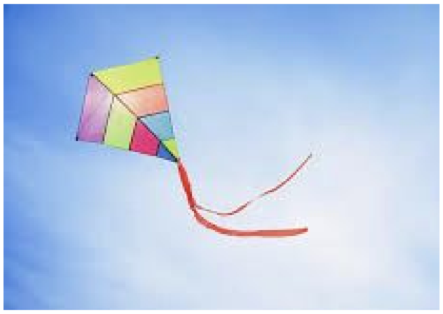
FIGURE 2: Average Loss of Cross-Entropy after 100 Steps for Different Learning Rates

Next, we moved on to poem generation. We tested different values of temperature which controls the randomness in the completion of words and phrases and top_k which controls the diversity, i.e. the number of words considered for generation at each step. We tried temperatures 0, 0.5, and 1 and top_k 0, 1, 10, 20, and 40. We tested 5 pictures/words for each combination: dog, kiss, school, New York, and inside, then chose the combination with the best poems. We could not find a metric that we liked to quantitatively choose the best poems due to the ambiguous nature of poetry, so we just read all of them and rated them ourselves.

Results and Discussion:

Here are some examples of our object detection model and its output.

IMAGE	OUTPUT, PROBABILITY

	dog : 0. 9992483854293823
	person : 0.9693329334259033 boat : 0.8722175359725952
	kite : 0.8208012580871582

We found the optimal learning rates for our GPT-2 model to be 0.00005 and 0.000005 from our average loss graph. The poem generation parameters that performed best were temperature = 1 and top_k = 40. From our poem readings, any/all temperatures below 1 tended to become too repetitive and would only repeat the same phrases over and over. Here are some of our final poems for our test words.

INPUT	SELECTED OUTPUT
Dog	was always there with me the sound of my footsteps was like my feet on the pavement the sounds of the snow were always there with me ...
kiss	and the other end of his nose and he kissed her back again and again and the next couple of hours had not passed for his tongue or his teeth but she moaned and said she did not know what he was doing but he had come to her and the rest is history for her ...
Boat and person	for my boat and a large boat it's a little sad but they were lucky enough that they kept me ...
New York	-i'll be back the place will become an endless parade for all that's left is a thousand years to die the past is no time to live the future is no place to hide i'm not going to sleep here was a time for it i'm not here you know it's a place for you i'm not here ...
Kite	“’’a friend once said how often do these people tell each other that one day the earth will be beautiful ''' but the other side of the

	room ‘’’ is it true how many times when the sun has risen for every thing is there a solar that it does not for every thing is there a planet or some kind of thing ''' he said so in that way we must take the tutelage of human nature ...
Peach banana	or blueberry salad but the only way to make the best of three very good foods is to combine them in one bowl and let them cool until completely absorbed into the mouth so the most important part of the best food is the two best so it's not even a question of whether or not that a good fruit salad is a good dessert i am an lover and i have often been told and heard that what should be eaten by us is

	a food that is one reason why so many children do not know that fruit salad is a special treat: you could eat three-ounce cake with cream it will take you a little but it is not the same because in our society it is very important not to eat food that would not spoil one's taste ...

Conclusion / Future Projects:

In this project, we performed and fine-tuned a simple approach for generating readable poetry from images. In the future, some interesting additions include adding a token with the author’s name at the beginning of each of their poems to be able to generate a certain style of poetry on demand, using a image captioning model instead of an object detection model, and using an LSTM for it’s memory properties for poetry generation.

Paper References:

[1] T.Lin, P.Goyal, R.Girshick, K.He, P.Dollar, Focal Loss for Dense Object Detection, (2017). https://arxiv.org/abs/1708.02002

[2] A.Radford, J.Wu, R.Child, D.Luan, D.Amodei, I.Sutskever: Language Models are Unsupervised Multitask Learners (2019). https://d4mucfpksywv.cloudfront.net/better-language-models/language_models_are_unsupervised_multitask_learners.pdf

[3]J.Lee, J.Hsiang, Patent Claim Generation by Fine-Tuning OpenAI GPT-2, (2019). https://arxiv.org/pdf/1907.02052.pdf

[4] A.Wang, K.Cho, BERT has a Mouth, and It Must Speak: BERT as a Markov Random Field Language Model, (2019). http://arxiv.org/abs/1902.04094 (accessed March1, 2019).

Blog References:

- **Image background**
  - M.Olafenwa, Object Detection with 10 Lines of Code, (2018), https://towardsdatascience.com/object-detection-with-10-lines-of-code-d6cb4d86f606
  - M.Olafenwa, Official English documentation for ImageAI, (2018), https://imageai.readthedocs.io/en/latest/
- **GPT-2 background**
  - A. Radford, J. Wu, D.Amodei, D.Amodei, J.Clark, M. Brundage, I.Sutskever, Better Language Models and Their Implications, (2019), https://openai.com/blog/better-language-models/
  - S.Lynn-Evans, The Transformer in PyTorch, (2018), https://blog.floydhub.com/the-transformer-in-pytorch/
  - N. Foong, Beginner’s Guide to Retrain GPT-2 (117M) to Generate Custom Text Content, (2019), https://medium.com/@ngwaifoong92/beginners-guide-to-retrain-gpt-2-117m-to-generate-custom-text-content-8bb5363d8b7f

Github References:

- **GPT-2 Model**
  - https://github.com/nshepperd/gpt-2

- https://github.com/researchmm/img2poem/tree/master/data