IMAGE CAPTIONING

Attempt 1

Hruday Kinhikar

Dataset

- Using Hugging Face coco dataset [1]
- Has 5 captions per image to cover many bases
- 200,000 images (25GB)
- Extensive coverage of everyday scenes - photographs

Model

ViT (vision transformer model):

- Transforms raw data into useful features

GPT-2:

- Tokenises features
- Generates captions after training on tokens

Sourced base code from [2]

Methodology

- Initial aim: get a working code and experiment
- Trained on a 1000 images – partially trained model
- Sourced a model trained on the entire coco dataset – fully trained model [3]
- Initially tested on coco test dataset
- Tested on client images so we have clients' model as a reference
- Feature extraction using vision transformer into sequence of embeddings
- GPT-2 tokenizer
- Configuration – appropriate token ids for special tokens like EOS and padding tokens
- Trained to minimise cross-entropy loss

Quick analysis of partially trained model

Training details:

- Trained locally on 16GB RAM laptop
- Preprocessing data took ~ 2 hours
- Original dataset: 25.1GB -> Processed dataset: 370GB
- Training the partially trained model took ~ 3 - 4 hours
- Parameters for both ViT and GPT2 were left as default

Model Performance

- Training loss: 0.3365
- Evaluation loss: 0.3013
- Usual range for cross entropy loss: 0.3 – 1.0 [4]
- ROUGE-1 : 21.84%
- Usual range is 40%-60% [6]
- ROUGE-2 : 2.25%
- Usual range is 20%-40% [6]

Testing with coco dataset

Fully trained model:

A soccer game with a player jumping to catch the ball

Partially trained model:

A man is holding a baseball bat in the air

Fully trained model:

A bathroom with a tub, sink, and mirror

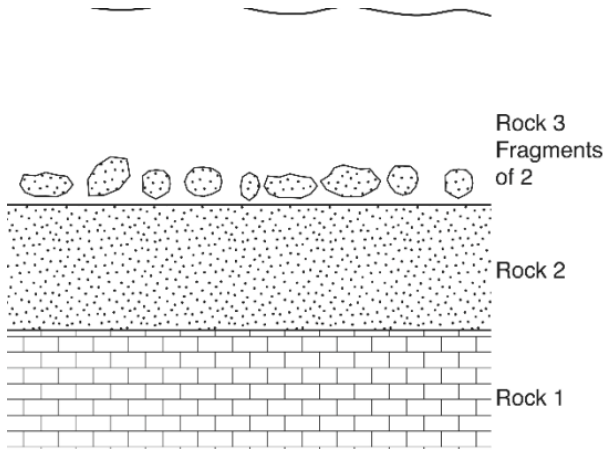
Partially trained model:

A bathroom with a shower and sink

Evaluation:

- Wherever partially trained model failed, it recognised general category of the image
- Partially trained model needs to be trained on a larger section of the dataset
- Need a way to measure how well our fully trained model behaves

Client images in which they succeeded



Fully trained model: a series of photographs showing a series of small, white objects

Partially trained model: A man standing in front of a large mirror

Client model: a diagram of a rock wall with a line of rocks on it

Fully trained model: a rocky cliff with a large rock wall

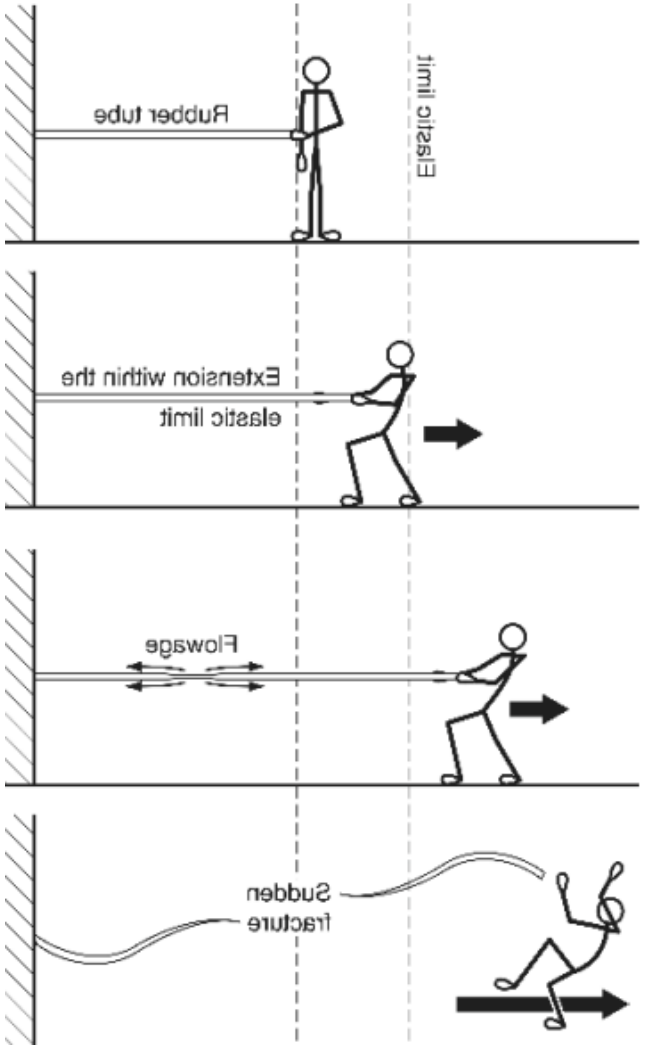
Partially trained model: A man riding a skateboard on a snowy mountain

Client model: there is a large rock formation with a hole in it

Evaluation:

- Fully trained model struggled on diagrams - need to train it on diagrams
- Both clients and our fully trained model did very well on landscape images

Client images in which they failed



Fully trained model: a collage of photos showing a woman cutting a pair of scissors

Partially trained model: A man standing in front of a large building with a clock

Client model: a diagram of a man doing a trick on a skateboard

Evaluation:

- All 3 identify a man/woman but fully trained model
- Skateboard and scissors are plausible readings if the resolution was lowered
- Need to find a way to use the text as context - OCR (Optical Character Recognition)
- Both partially and fully trained models failed to identify a tree diagram – dataset needs to include diagrams

Overall evaluation: with the aim of improving performance

- Need to diversify training data to include images of a diagrams
- We want data with more descriptive captions that aim to describe everything in the image, so we'll use [5] dataset instead
- Need to incorporate OCR to include context based off the text in the images
- Explore any models which are more 'powerful' than GPT2 when it comes to learning from images

Next steps

- Put a bigger focus on datasets – specialise datasets, explore interesting metrics etc. (currently looking into satellite images for ship detection)
- Explore how Kaggle users work with visual data in terms of sorting, splitting, cleaning etc.
- Need to incorporate a reinforcement learning loop
- Step project down to image classification for implementation of RL and complicate as needed

References

[1] huggingface.co. (n.d.). nlpconnect/vit-gpt2-image-captioning · Hugging Face. [online] Available at: https://huggingface.co/nlpconnect/vit-gpt2-image-captioning

[2] Kumar, A. (2022). The Illustrated Image Captioning using transformers. [online] Ankur | NLP Enthusiast. Available at: https://ankur3107.github.io/blogs/the-illustrated-imagecaptioning-using-transformers/

[3] huggingface.co. (n.d.). nlpconnect/vit-gpt2-image-captioning · Hugging Face. [online] Available at: https://huggingface.co/nlpconnect/vit-gpt2-image-captioning.

[4] Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhutdinov, R., Zemel, R., & Bengio, Y. (2015). Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. arXiv preprint arXiv:1502.03044

[5] google.github.io. (n.d.). DOCCI. [online] Available at: https://google.github.io/docci/ [Accessed 22 Jun. 2024].

[6] FreeCodeCamp. (2024). An intro to ROUGE, and how to use it to evaluate summaries. Retrieved from FreeCodeCamp