Reverse engineering of Stable Diffusion 2.0 - generating prompt based on image

Author: Paweł Kalisz
Thesis supervisor: Jacek Komorowski PhD

Warsaw
03.03.2024

Wydział Elektroniki i Technik Informacyjnych
Politechnika Warszawska

Objectives

- The main goal of the thesis is to develop deep neural network model predicting prompt used to generate given image generated by Stable Diffusion 2.0

- The assumption was to train model only on the images generated by Stable Diffusion 2.0

- Two models were implemented, trained and evaluated
  1. Model based on Encoder CNN – decoder LSTM with Bahdanau attention
  2. Model based on Transformer architecture with CNN for features extraction

Expected result

Both models were trained on pairs consisting of generated image and prompt used for its generation

Model input (image generated by Stable Diffusion 2.0):

Model target (prompt used to generate the image):

facing the dark star with a sword in hand galactic nebular astral realm sacred journey in oil painting trending on artstation award winning emotional highly detailed surrealist art

Data set and data processing

Models were trained, validated and tested on the data set DiffusionDB 2M consisting of 2 milion pairs of prompts and generated images. Because of limited memory capacity of Colab platform, the data set was reduced to images with high to width ratio equal to 1:

- 1,084,245 images and
- 884,825 unique prompts

Images potentially harmul for the model were identified based on metadata and dropped:

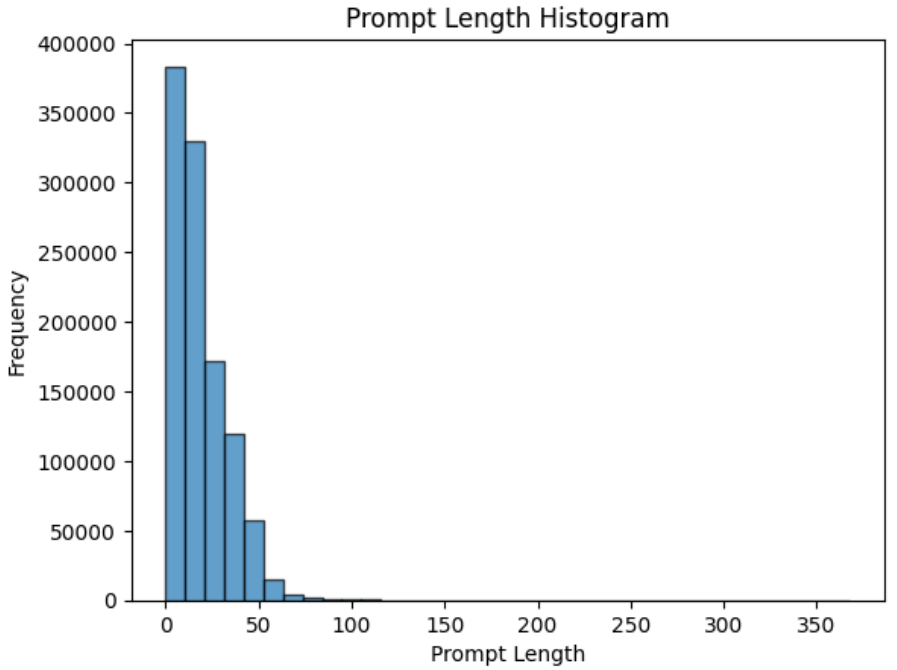
- NSTW = 2: images with NSTW (Not Safe For Work) equal to 2 were blurred by Stable Diffusion
- CFG < 0: CFG is a Stable Diffusion hyperparameter defining how much generated image will be similar to the given prompt. Value below 0 means, that image will be opposite of prompt
- Step < 10: Low value of this hyperparameter can cause image to be blurred

Images were resized to resulotion 224 x 224

Data set was divided to training (90%), validation (5%) and test set (5%)

Data set and data processing

- Prompts were tokenized by tokenizer „punkt” available in NLTK library removing interpunction from text. Tokens containing numers were also removed.
    - **Before tokenization**: "george washington by kehinde wiley and rembrandt : 5. photorealistic facial features, portrait, detailed face, ultra high details, large, close - up : 1."
    - **After tokenization**: "george", "washington", "by", "kehinde", "wiley", "and", "rembrandt", "photorealistic", "facial", "features", "portrait", "detailed", "face", "ultra", "high", "details", "large", "close", "up"
- Tokenized text was cut to 30 tokens based on prompts length histogram



Images augmentation:

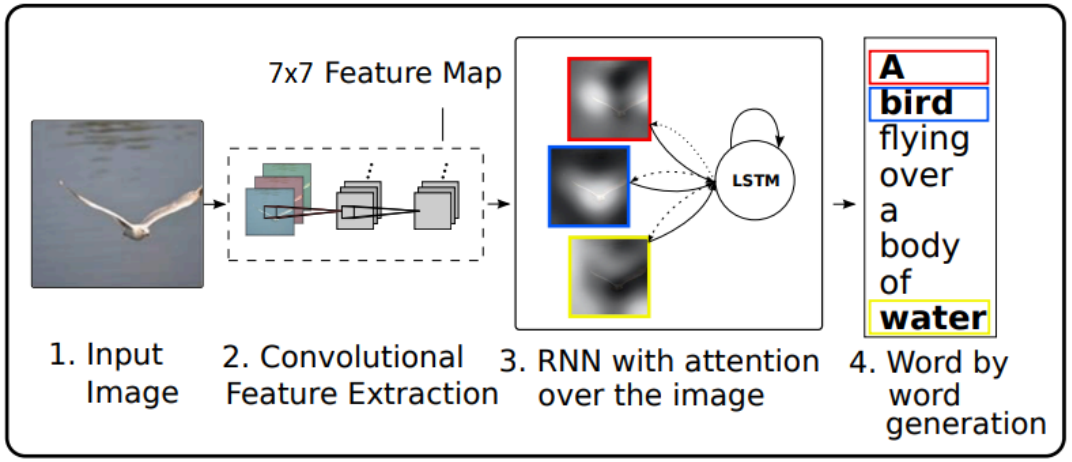
- Random Rotation
- Random Horizontal Flip
- Random Erasing

Encoder CNN – Decoder LSTM

- Encoder is based on pre-trained ResNet50 with frozen weights removed top two layers giving features map of 49 x 2048
- Decoder is based on LSTM cells with Bahdanau attention
- Model was trained for 50 epochs, but based on loss curves checkpoint 27 was chosen

Encoder CNN – Decoder LSTM

```
==========================================================================
Layer (type:depth-idx)                                   Param #
==========================================================================
EncoderDecoder                                           --
├─BackboneCNN: 1-1                                       --
│    └─Sequential: 2-1                                   --
│    │    └─Conv2d: 3-1                                  (9,408)
│    │    └─BatchNorm2d: 3-2                             (128)
│    │    └─ReLU: 3-3                                    --
│    │    └─MaxPool2d: 3-4                               --
│    │    └─Sequential: 3-5                              (215,808)
│    │    └─Sequential: 3-6                              (1,219,584)
│    │    └─Sequential: 3-7                              (7,098,368)
│    │    └─Sequential: 3-8                              (14,964,736)
├─DecoderRNN: 1-2                                        --
│    └─Vocab: 2-2                                        --
│    └─Embedding: 2-3                                    23,891,968
│    └─Attention: 2-4                                    --
│    │    └─Linear: 3-9                                  131,328
│    │    └─Linear: 3-10                                 524,544
│    │    └─Linear: 3-11                                 257
│    └─Linear: 2-5                                       1,049,088
│    └─Linear: 2-6                                       1,049,088
│    └─LSTMCell: 2-7                                     6,295,552
│    └─Linear: 2-8                                       1,050,624
│    └─Linear: 2-9                                       23,938,632
│    └─Dropout: 2-10                                     --
==========================================================================
Total params: 81,439,113
Trainable params: 57,931,081
Non-trainable params: 23,508,032
==========================================================================
```

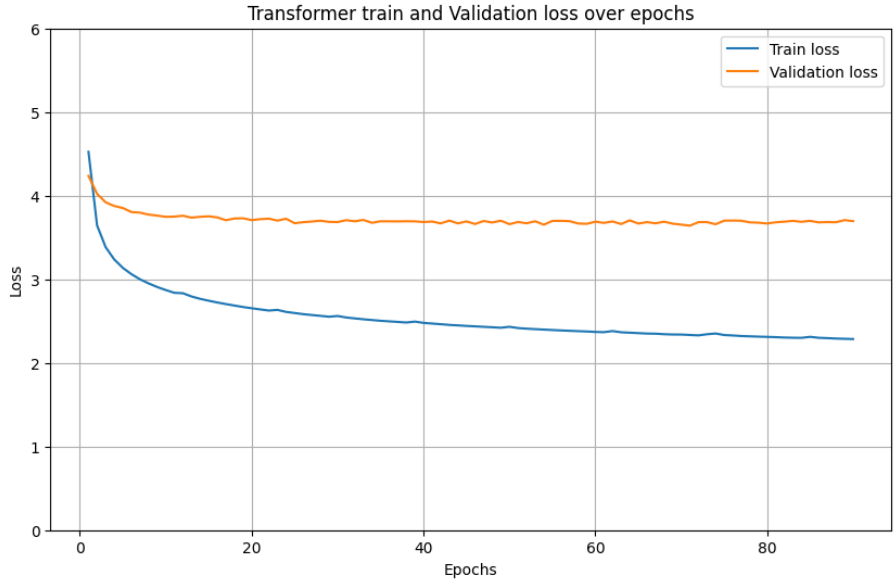


Model based on Transformer

- Encoder:
  1. Pre-trained ResNet50 with frozen weights extracts features from image with shape of 49 x 2048
  2. Extracted features are passed to standard Transformer positional encoding like 49 tokens with 2048 dimensions
  3. Next features are passed to 2 Transformer Encoder blocks with 4 attention heads

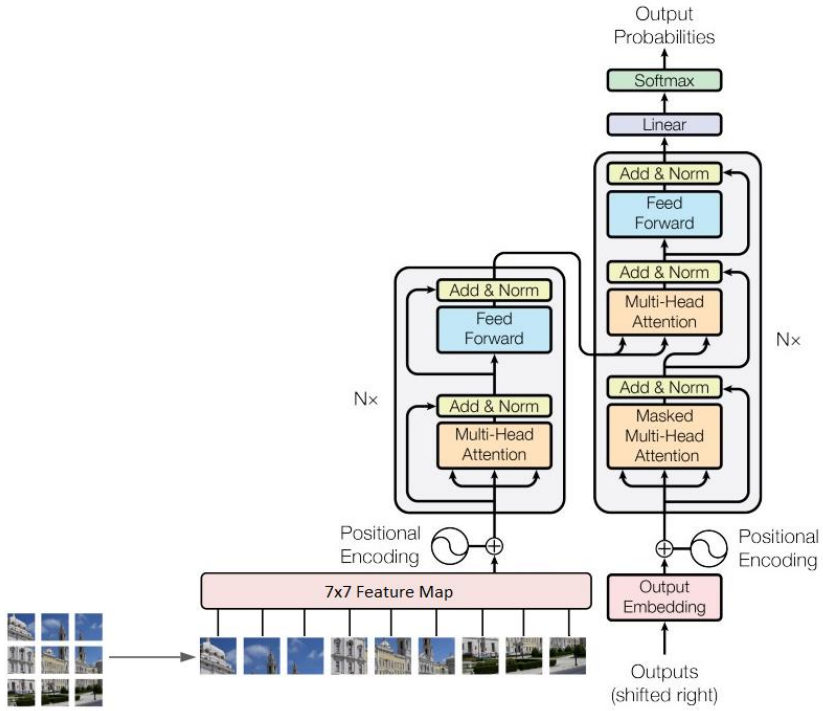
- Decoder:
  1. Tokenized text is firstly embedded to 512 dimensions with max sequence size of 30
  2. Embedded text is then positionally encoded
  3. In the next step processed text is passed to 2 standard Transformer Decoder blocks with 4 attention heads

- Model was trained for 90 epochs, but based on loss curves checkpoint 71 was chosen
- Training took 180 hours on NVIDIA V100



Model based on Transformer

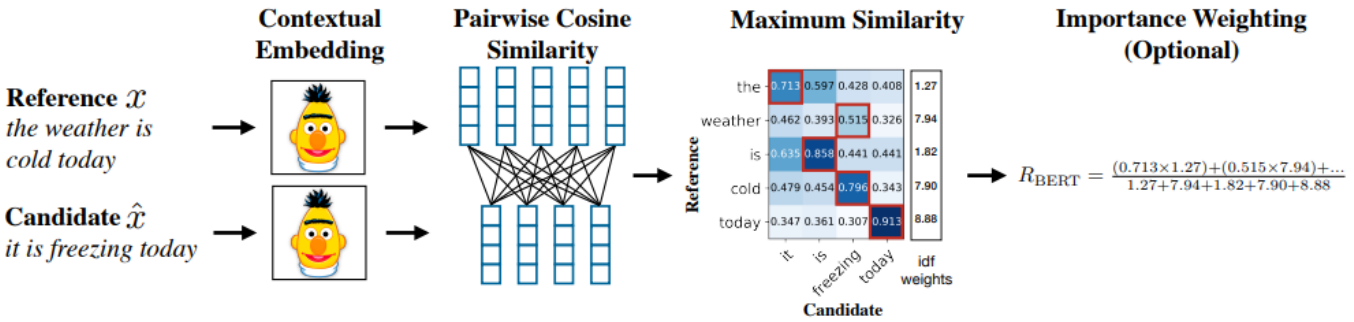
```
==========================================================================================
Layer (type:depth-idx)                                   Param #
==========================================================================================
Transformer                                              --
├─Encoder: 1-1                                           --
│    └─BackboneCNN: 2-1                                  --
│    │    └─Sequential: 3-1                              (23,508,032)
│    └─PositionalEncoding: 2-2                           --
│    │    └─Dropout: 3-2                                 --
│    └─Sequential: 2-3                                   --
│    │    └─EncoderBlock: 3-3                            34,099,392
│    │    └─EncoderBlock: 3-4                            34,099,392
│    └─LayerNorm: 2-4                                    4,096
├─Decoder: 1-2                                           --
│    └─Embedding: 2-5                                    23,891,968
│    └─PositionalEncoding: 2-6                           --
│    │    └─Dropout: 3-5                                 --
│    └─Sequential: 2-7                                   --
│    │    └─DecoderBlock: 3-6                            2,562,944
│    │    └─DecoderBlock: 3-7                            2,562,944
│    └─LayerNorm: 2-8                                    1,024
│    └─Linear: 2-9                                       23,938,632
==========================================================================================
Total params: 144,668,424
Trainable params: 121,160,392
Non-trainable params: 23,508,032
==========================================================================================
```



Evaluation

Metrics used in machine translation were used for models evaluation:

- BLEU
- METEOR
- BERT Score



Model	BLEU	METEOR	BERT Score F1
LSTM	1.1	8.1	44.1
Transformer	5.0	14.1	51.6

Conclusions

- Model based on Transformer performs way better than the one based on CNN-LSTM. The latter one often struggles to generate a coherent text
- Even though model based on Transformer often describes well content of the image, it performs much worse than state-of-the-art models trained and evaluated on data sets like Flickr8k or COCO
- Unsatisfactory performance is mainly caused by loss of information. Often a large part of the prompt is not reflected in generated image making prediction of the missing parts impossible. On the other hand Stable Diffusion 2.0 can generate objects not described in the prompt

Models demonstration

True prompt: purple cat sitting in a lunar lander on the surface of the moon in the style of greg Rutkowski
Transformer: a blue and white cat sitting on the moon with stars in the background
CNN-LSTM: a black hole in the shape of a a

True prompt: five monkeys in dresses dancing in the jungle in the style of renoir
Transformer: a group of people dancing in a jungle party in the style of renoir
CNN-LSTM: a painting of a a

Models demonstration

True prompt: indiana jones riding a cow with a magic wand in his hand on the street of new york
Transformer: a cow riding a bull in new york city
CNN-LSTM: a photo of a a

True prompt: banana shaped mouse in the fantasy forest in the style of rembrandt
Transformer: maus in forest by rivuletpaper rivuletpaper art mouse guard by lily seika jones rivuletpaper art top cinematic lighting cinematic mood very detailed shot in canon high detail mood
CNN-LSTM: a painting of a a

Models demonstration

True prompt: walter white cooking dinner for his family at the swimming pool party
Transformer: walter white cooking spaghetti in a diner
CNN-LSTM: a still of a in a

True prompt: darth vader ordering beer in the bar on tatooine highly detailed
Transformer: darth vader drinking beer at starbucks cinematic lighting
CNN-LSTM: a still of a in a

Models demonstration

True prompt: a fish running a marathon on the beach highly detailed
Transformer: a beautiful painting of a trout swimming in the ocean
CNN-LSTM: a fish swimming in a pool of water

Thank you!