img2capt: Deep Learning Image Caption Generator

Matan Sheskin, ID: 304826811

May 2021

Final Project for Deep Learning course, IDC, Semester 2021A

Contents

1 Introduction

"One picture is worth a thousand words", but what about the rest of the words? The visual world and the linguistic world use different atomic components to convey information: Images and words. Images directly reveal all content to a willing viewer, and do not impose a single agreed-upon method to interaction by the viewer: Wherever you look, and however you look, regardless of where you looked before, the image remains an image - unforgiving, blunt, honest. Words are more elusive: They require a unique method of interaction from a reader who wishes to distill information from them. Words form a road, and a reader must go through the road, step by step, word by word, to learn it.

As a Computer Science undergraduate student, the study of words and the study of images seemed separate. Research on words is done in Natural Language Processing (NLP), while research on images is done with Computer Vision. Separation seemed intuitive because of the serialization factor. But while the visual world and the linguistic world both choose different ways to represent and convey information, they both represent and convey information through components, and component composition. Such composition adds layers of complexity, meaning, and definiteness, that allow information to pass. In my final project for neural networks, I wanted to find a bridge between the two worlds.

Tasks that have footing in both worlds, such as image classification or object recognition have been a main focus in the computer vision community [18]. However, just like a sentence has an added layer of meaning that builds upon existing interaction between words, an image has an added layer of meaning that builds upon the interaction between objects contained in it. Thus, a description of an image must express how objects relate to each other, which is a delicate semantic relation that carries with it large variation, complexity, uniqueness, and importance. Moreover, novel descriptions have to be possible. Natural languages such as English, that possess high expressive power, must then be considered to form a bridge between words and images [12].

Before considering image-to-text translation, it might be beneficial to look into text-to-text translation. In machine translation, the task is to transform a sentence S written in a source language, into its translation, T, in the target language by maximizing $p(T|S)$. Past attempts to tackle this problem involved decomposing the task into smaller tasks: First decompose a sentence into words, then translating words individually, then align them, reorder them, and so on [2]. These steps were computationally costly, since they tried to restore lost sequential information bottom-up, imposed by the initial decomposition. Incorporating Recurrent Neural Network (RNNs) that maintain sequential information simplified the process: An "encoder" RNN reads the source sentence and transforms it into a rich fixed-length vector representation that contains order information. This vector is then used as the initial hidden state of a "decoder" RNN that generates the target sequence. Such networks maintained a high level of performance [4, 21, 1]

Influenced by these advances in text-to-text translation, Vinyals and his team proposed a similar implementation for image-to-text translation [22]. They offered to take the previous model and replace the encoder RNN with a deep convolution neural network (CNN). CNNs can reproduce a rich representation of the input image by embedding it to a fixed-length vector. This representation can then be used for machine vision tasks [19]. Hence, it can be used as an image "encoder", by first pre-training it for an image classification task, and then using the last hidden layer as an input to the RNN decoder that generates sentences. The model suggested was called Neural Image Caption (NIC). NICE takes an image I as input, and is trained to maximize the likelihood $p(S|I)$ of producing a target sequence of words $S = s_1, s_2, ...$, where each word, $s_t \in S$, comes from a given dictionary, that describes the image adequately. This is an analogous process to machine translation, but with the added condition that sentences must be related to an underlying non-textual modality, such as an image. My implementation is based on this paper, alongside available implementations in GitHub [20, 17].

1.1 Related Works

There are other ways to perform caption generation. Kiros [10] proposed a multimodal log-bilinear feed-forward model that generalizes the task beyond images, and into any modal dependant translation requirement from which numerical representation can be extracted. By doing so, Kiros attempted to encode multimodal information which represents context into a unified multimodal space, an activity known as co-embedding. Kiros also proposed a multimodal neural language model that performs this [11]. However, to balance the complexity added from dealing with multimodal information, the final textual descriptions had to be structural. Such concession reduces the ability of the language model to create novel descriptions.

One factor contributing to the inability to generate novel descriptions is the loss of semantic information. Without semantic information, the model gives more frequent words precedence over similar, less frequent, words. This, in turn, causes less frequent words to become used even less, contributing to a state of sparsity in used vocabulary, that reduces the ability to generate novel descriptions [9]. To deal with this, Mikolov and Zweig [16] used RNN with LSTM, to compute a context vector of a sentence. This information, when used as an added input to the model, allowed it to consider more similarities between words (topic modelling) from before, with little overhead.

This work echoes the idea that an underlying latent semantic structure to which words belong, might better describe concepts than the words themselves, which lies in the heart of Latent Semantic Analysis [5]. A unified semantic space appeals not just due to is ability to handle a large vocabulary, but because it provides a tool to compress large variation in general, in all multimodal forms. Following this line of thought, Mao and his team [15] substituted the feed-forward component in Kiros's work with with a RNN, to build a multimodal RNN model. Hence, RNN was becoming a prominent figure in the world of modal-to-text translation.

The most common challenge in designing and training RNNs is vanishing and exploding gradients, for which a particular form of recurrent net called LSTM was developed [8]. LSTM was then also applied to image-to-text translation [4, 21]. The added performance boost supplied by LSTM allowed researches to spread outside of static images, and into videos [6].

A potential issue with encoder-decoder approach is that a neural network needs to be able to compress all the necessary information of source sentence into a fixed-length vector. But this suggests that performance of a basic encoder-decoder on sentences deteriorates rapidly as the length of an input sentence increases, a result shown by Cho et al. [3]. To deal with this, attention was introduced as an extension to the encoder-decoder model: Instead of trying to encode a whole input sentence into a single fixed-length vector, the model first encodes input sentences into a sequence of vectors and then adaptively chooses a subset of these vectors while decoding the translation [24].

2 Implementation

2.1 Design

2.1.1 Platform

The model was built using PyTorch. It was trained on Google Colab, so that GPUs are available for faster training. Images were downloaded from the V6 Google Open Images Dataset.

2.1.2 Architecture

A complete architecture of the image captioning model: The CNN encoder is a residual network (ResNet) 152 pre-trained on the ImageNet database. Its role is to extract features from images and passes these to the RNN LSTM encoder. The decoder finds a sequence of words, according to time steps, that describe the image, until the 'end' token is met.

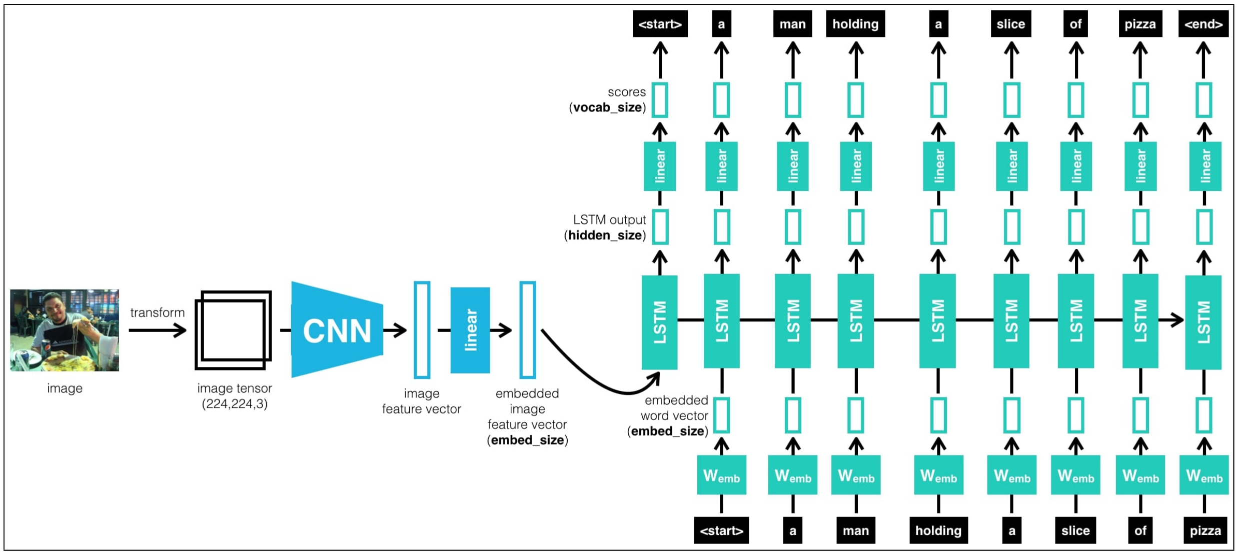


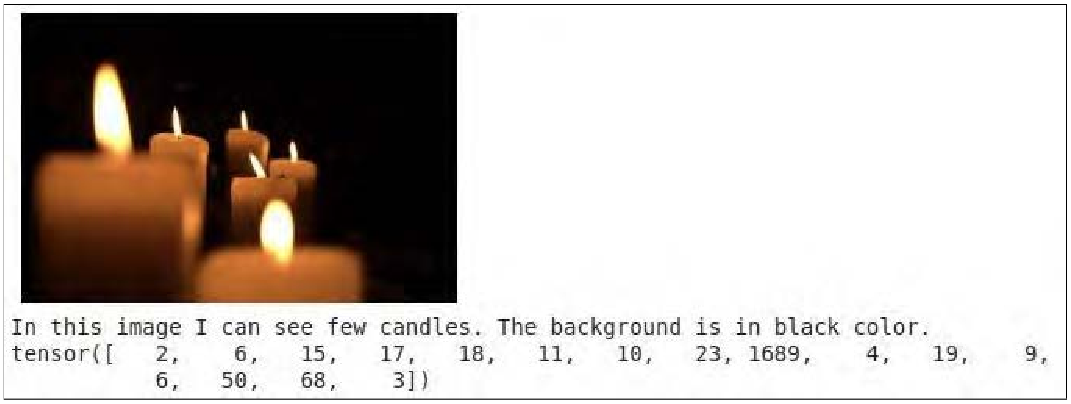
Figure 1: Complete model architecture. LSTM model is described in unrolled form.

2.2 Procedure

To build the caption generator, the following procedure was performed:

2.2.1 Preprocessing

We accept captions as input. Each unique word is then identified by a unique ID (one-hot encoding). This translation helps in embedding, which is managed by the decoder. We also define actionable tokens such as 'start' and 'end', that help our model know when to stop generating predictions. We also use the 'pad' token to help align caption sizes, and ensure all embeddings have the same length. Our output is integer tokens of same length, padded with auxiliary tokens.



```
In this image I can see few candles. The background is in black color.
tensor([   2,    6,   15,   17,   18,   11,   10,   23, 1689,    4,   19,    9,
           6,   50,   68,    3])
```

Figure 2: Words are uniquely encoded to integers.

2.2.2 Encoding

As input, we get an image. The truncated ResNet-152 encoder extracts features from this image. These features are then reshaped and normalized in preparation for the decoder. Our output is normalized image features.

2.2.3 Decoding

As input, we get image features prepared by the CNN encoder. We also get corresponding captions as integer tokens. First, we convert caption tokens into embeddings. Then, we concatenate corresponding image features with our embeddings. Notice that time steps, denoted L, is thus elongated by 1 after concatenation.

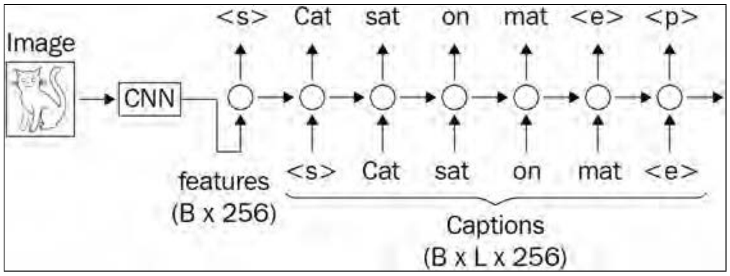
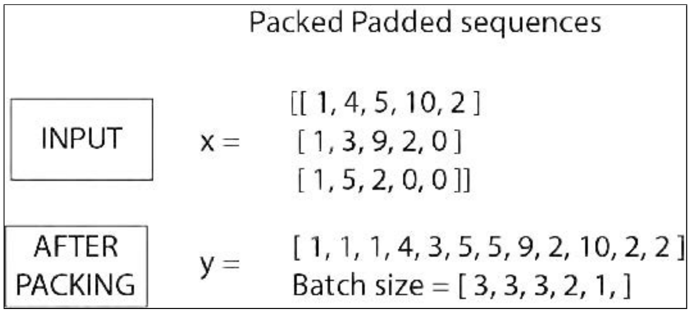


Figure 3: Learned Image features are concatenated with captions.

We then pack the concatenated sequences to remove padded tokens. Packing helps in making sure that unrolling does not occur at time steps where padding is present. As a result, RNN computations are more efficient. In the following illustration, three sentences are presented, encoded according to word indices. Index 0 represents the padding index. The batch size contains the number of non-padded elements for each index in the sentence. Since there is only one sentence where the last index in the sentence is a non-padded element, the last index element in batch size is 1.



Packed Padded sequences

INPUT

x = [[1, 4, 5, 10, 2]
[1, 3, 9, 2, 0]
[1, 5, 2, 0, 0]]

AFTER PACKING

y = [1, 1, 1, 4, 3, 5, 5, 9, 2, 10, 2, 2]
Batch size = [3, 3, 3, 2, 1,]

Figure 4: Packing Illustration.

Packed sequences are then sent to LSTM as concatenated packed embeddings. After LSTM, a linear layer flattens the predictions. A normalization software layer then produces outputs.

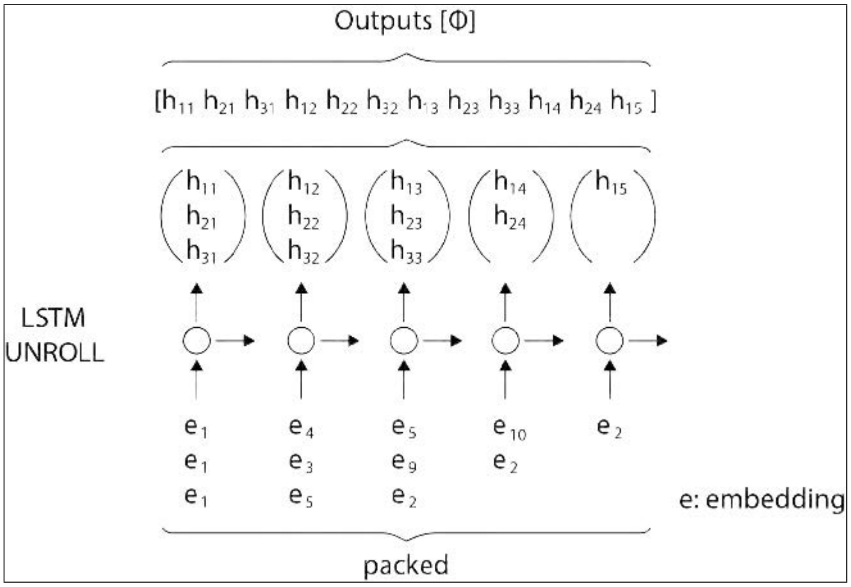


Figure 5: Packed padding are fed into the LSTM.

2.3 Training

The LSTM model is trained to predict each word of the caption sentence after it has seen the image, and all preceding words. In other words, it calculates $p(S_t|I, S_0, ..., S_{t-1})$, where I denotes the image, and S_j denotes the word corresponding to index $0 \leq j$. All LSTMs share the same parameters and output of the immediate LSTM temporal predecessor.

The chosen **loss function** was Cross-Entropy loss, the same as in Vinyals's study [22]. Following Goodfellow [7], to account for decay and avoid overfitting, L2 regularization is used. Nevertheless, since we are using an adaptive algorithm, care must be taken so that calculated gradients are not affected by the added regularization term. Hence, the **AdamW optimizer** was used [14]. The initial **adaptive learning rate** the optimizer received was 0.0004.

Overall, the model was trained on randomly chosen **50,000 images**, of which 95% were randomly assigned to the training set. The rest were assigned to test set. The number of epochs was 10. Total training time took 4 hours. Due to technical difficulties (browser tab kept crashing), I was not able to train the model for longer.

3 Results

3.1 Loss variation

Model evaluation was carried after training was completed. The following table shows the variation of the training and validation loss over increasing epochs. Performance is increasing. Other ways to measure performance, such as BLEU-1 scores were not implemented.

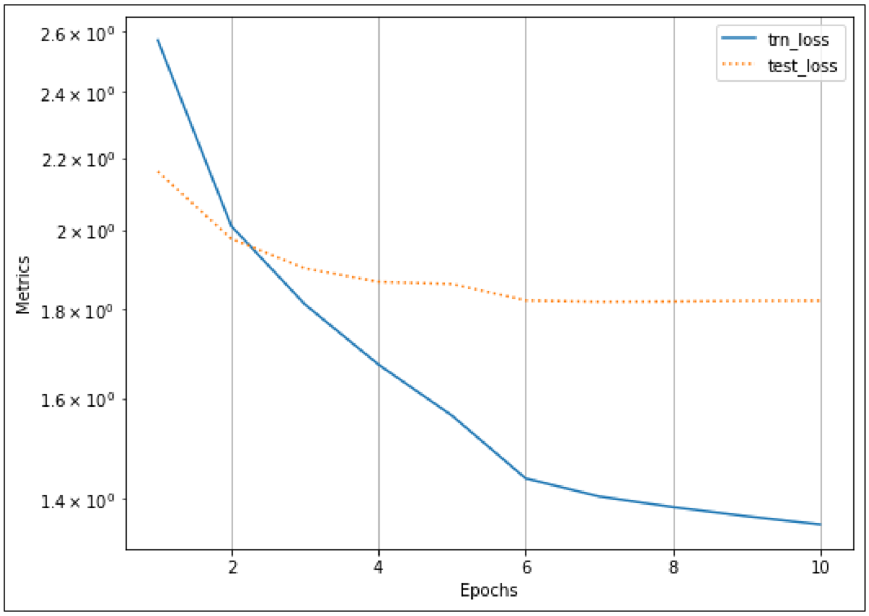


Figure 6: Variation of training and validation loss over increasing epochs

3.2 Output caption analysis

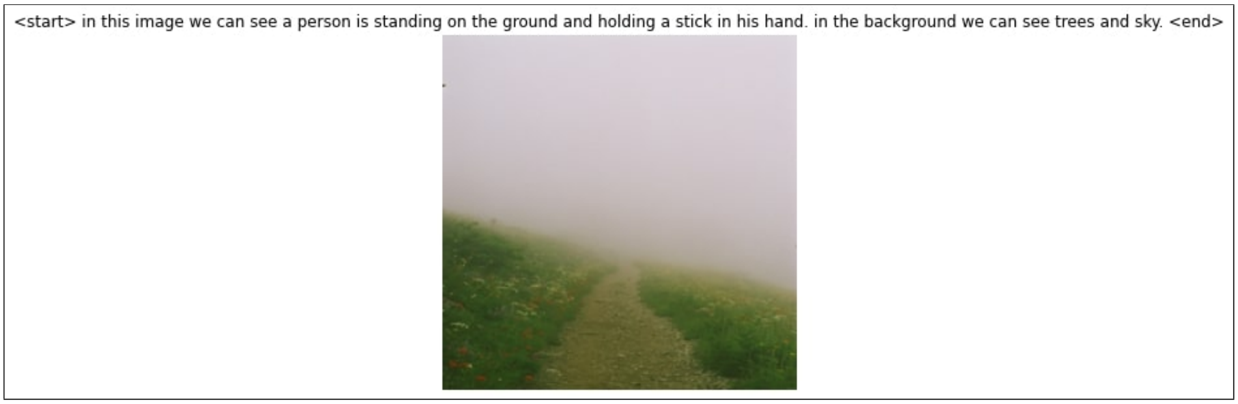
Five images are presented alongside their generated captions.

Regarding syntax, all captions perform well: They make sense, both as individual components, and as a sequential unit of text that also includes punctuation.

In regard to their connection to the depicted image, there is variation:

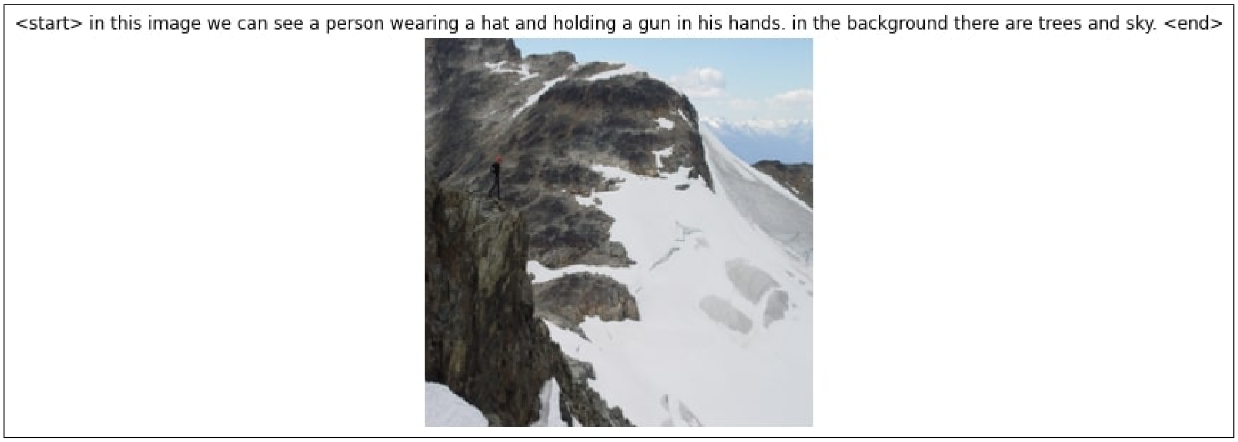
3.2.1 Captions 1 and 5

These images perform poorly. While they manage to identify the sky, non existing elements are also described: In image 1, the caption depicts a non existing person holding a stick. In image 5, the caption successfully recognizes a single person, but it also adds a non existing gun in his hand. Interestingly, image 5 also successfully predicted that the person is a man, by using the word 'his' to describe his hands.



<start> in this image we can see a person is standing on the ground and holding a stick in his hand. in the background we can see trees and sky. <end>

Figure 7: Generated Caption 1



<start> in this image we can see a person wearing a hat and holding a gun in his hands. in the background there are trees and sky. <end>

Figure 8: Caption 5

3.2.2 Caption 2

Caption 2 performs adequately. It succeeds in describing all elements in the image, even less apparent elements that require zooming in.

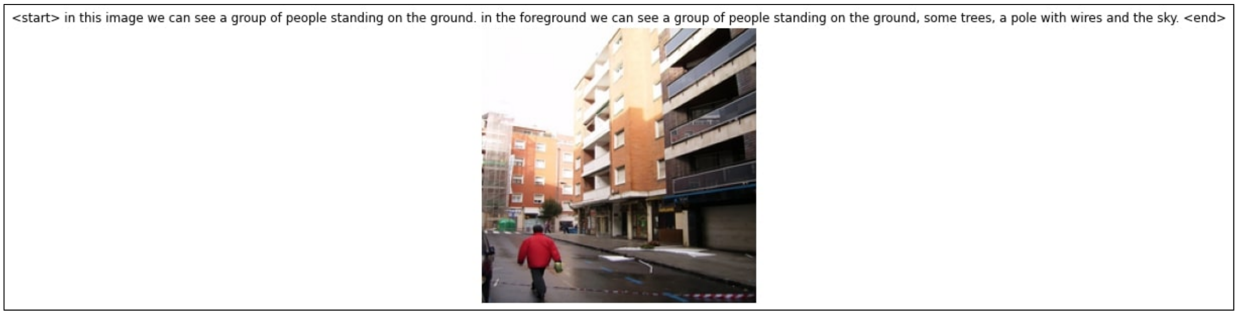
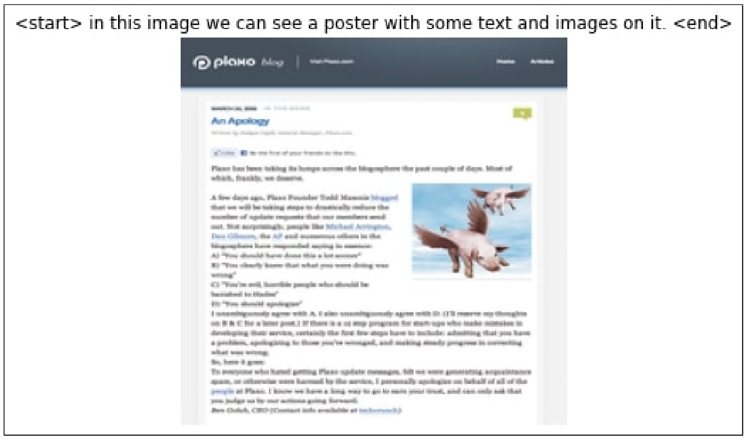

<start> in this image we can see a group of people standing on the ground. in the foreground we can see a group of people standing on the ground, some trees, a pole with wires and the sky. <end>

Figure 9: Caption 2

3.2.3 Caption 3

Caption 3 misclassifies a webpage for a poster. Nevertheless, it does correctly recognize text and images, and it does not add non-existing elements.


<start> in this image we can see a poster with some text and images on it. <end>

plaxo blog | [illegible]

[illegible]

An Apology

[illegible]

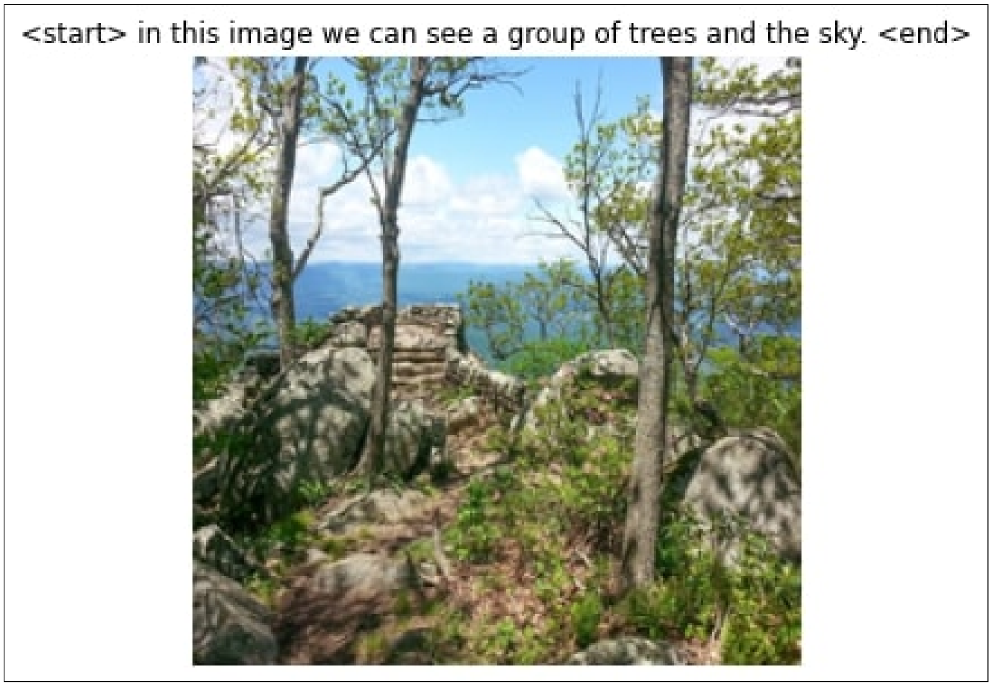
Plaxo has been taking its lumps across the blogosphere the past couple of days. Most of which, frankly, we deserve.

A few days ago, Plaxo Founder Todd Masonis blogged that we will be taking steps to drastically reduce the number of update requests that our members send out. Not surprisingly, people like Michael Arrington, Dan Gillmor, the AP and numerous others in the blogosphere have responded saying in essence:
A) "You should have done this a lot sooner"
B) "You clearly knew that what you were doing was wrong"
C) "You're evil, horrible people who should be banished to Hades"
D) "You should apologize"
I unambiguously agree with A. I also unambiguously agree with D. (I'll reserve my thoughts on B & C for a later post.) If there is a 12 step program for start-ups who make mistakes in developing their service, certainly the first few steps have to include: admitting that you have a problem, apologizing to those you've wronged, and making steady progress in correcting what was wrong.
So, here it goes:
To everyone who hated getting Plaxo update messages, felt we were generating acquaintance spam, or otherwise were harmed by the service, I personally apologize on behalf of all of the people at Plaxo. I know we have a long way to go to earn your trust, and can only ask that you judge us by our actions going forward.
Ben Golub, CEO (Contact info available at [illegible])

Figure 10: Caption 3

3.2.4 Caption 4

Caption 4 offers the shortest description of all captions. Nevertheless, it does not depict non-existing elements, and there is beauty in brevity.



<start> in this image we can see a group of trees and the sky. <end>

Figure 11: Caption 4

3.2.5 Conclusion

Overall, the descriptions seem shallow, focusing more on object detection and technical composition into a language model. That said, the captions seem to possess some descriptive ability.

4 Discussion

Our model provides in supplying partial descriptions to images. Nevertheless, our results fail to provide an evaluation on model performance. This evaluation is missing both internally, with regards to hyperparameter tuning, and externally, in regards to comparing the model results to results from other caption generators, and across multiple datasets. It should also be noted that failing to run the model in a functioning environment infused much overhead to each execution. That resulted in ultimately presenting a timid version of our model, that does not live up to its potential. Moreover, the main concern Vinyals et al. [22] mentioned was overfitting. This model does not supply adequate tools to gauge overfitting.

Another weakness of the model, which is common to all RNN models, lies in its adherence to use a training strategy that relies heavily on the maximum likelihood principle. In RNN literature, this form of training is known as teacher forcing [23], where the RNN model is guided strictly by ground-truths. This poses another limitation on the model, beyond overfitting: If the model is accustomed to use mainly ground-truths, its capability to generate novel captions is hindered, which will hinder the model's ability to adequately describe previously unseen images. Thus, model conditioning affects model performance. To deal with this challenge, Lamb and his team [13] offered a new algorithm for training RNNs, called Professor Forcing. In it, RNNs are encouraged to generate novel descriptions that are fed into subsequent layers arbitrarily instead of ground-truths. This conditions the RNN for creativity.

In conclusion, the world of words and the world of images share a composite complex relation. Looking back at my initial goal for the project, attempting to build a bridge between the world of words and the world of images seems overly ambitious. Nonetheless, this project took me on a journey that challenged my conception of what information is, and presented how rich the act of translation can be. I find myself consider the idea of latent semantic structures, and ask whether translation can, by itself, be regarded as such structure, composed not by passing modalities of information, but by similarities between them, and the people who use them.

5 Code

GitHub repository.

Bibliography

[1] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[2] P. F. Brown, S. A. Della Pietra, V. J. Della Pietra, and R. L. Mercer. The mathematics of statistical machine translation: Parameter estimation. *Computational linguistics*, 19(2):263–311, 1993.

[3] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*, 2014.

[4] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[5] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391–407, 1990.

[6] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2625–2634, 2015.

[7] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016.

[8] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[9] N. Kalchbrenner and P. Blunsom. Recurrent continuous translation models. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1700–1709, 2013.

[10] R. Kiros, R. Salakhutdinov, and R. Zemel. Multimodal neural language models. In *International conference on machine learning*, pages 595–603. PMLR, 2014.

[11] R. Kiros, R. Salakhutdinov, and R. S. Zemel. Unifying visual-semantic embeddings with multimodal neural language models. *arXiv preprint arXiv:1411.2539*, 2014.

[12] G. Kulkarni, V. Premraj, V. Ordonez, S. Dhar, S. Li, Y. Choi, A. C. Berg, and T. L. Berg. Babytalk: Understanding and generating simple image descriptions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(12):2891–2903, 2013.

[13] A. Lamb, A. Goyal, Y. Zhang, S. Zhang, A. Courville, and Y. Bengio. Professor forcing: A new algorithm for training recurrent networks. *arXiv preprint arXiv:1610.09038*, 2016.

[14] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[15] J. Mao, W. Xu, Y. Yang, J. Wang, Z. Huang, and A. Yuille. Deep captioning with multimodal recurrent neural networks (m-rnn). *arXiv preprint arXiv:1412.6632*, 2014.

[16] T. Mikolov and G. Zweig. Context dependent recurrent neural network language model. In *2012 IEEE Spoken Language Technology Workshop (SLT)*, pages 234–239. IEEE, 2012.

[17] S. Neerav. Image-captioning. `https://github.com/neerav47/Image-Captioning`, 2019.

[18] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

[19] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*, 2013.

[20] S. Siddharth. Image-captioning. `https://github.com/siddsrivastava/Image-captioning`, 2019.

[21] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. *arXiv preprint arXiv:1409.3215*, 2014.

[22] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3156–3164, 2015.

[23] R. J. Williams and D. Zipser. A learning algorithm for continually running fully recurrent neural networks. *Neural computation*, 1(2):270–280, 1989.

[24] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057. PMLR, 2015.