A Modern BERT Approach to Image Caption Generation

Abstract

The task of image caption generation requires an accurate understanding of visual objects and textual expression of that scene understanding. Recently, several prominent methods have been developed to resolve the task of generating image captions, inspired by the invention of attention-based approaches and pre-trained language models. In this paper, we implemented our baseline model of the captioning task, the encoder-decoder network integrated with soft-attention concepts, as a replication of [8]. As an extended approach to the captioning task, we also integrated pre-trained word embeddings of BERT [2] into this baseline model to enhance the performance of the baseline soft-attention model. We trained and evaluated these two captioning models on the Microsoft COCO 2014 train/validation images. During our experiments, we confirmed that the soft-attention model integrated with BERT's pre-trained embeddings outperforms the baseline soft-attention model by reducing cross-entropy loss of the baseline model in half and increasing BLEU scores significantly.

1. Introduction

Image caption generation is a task of detecting objects in computer vision (CV) as well as generating an accurate description of the detected objects with a neural language model in natural language processing (NLP). In the past few years, an advent of attention-based Transformer language models as well as state-of-art object detection has changed the approach to captioning tasks and thus enhanced the quality of generated captions. Therefore, we examine the modern approaches to generating image captions that use encoder-decoder models that are popularly utilized in both CV and NLP fields. To be specific, we implement encoder-decoder models with soft attention [8] as our baseline model.

Also, another objective of the paper is to extend the baseline model by incorporating pre-trained BERT word embeddings into the baseline model's decoder. By doing so, we enhance the baseline model's performance. The contributions of this paper are as follows:

- First, we implement the encoder-decoder model with soft-attention mechanism as the baseline model, as presented in [8].
- Second, we incorporate pre-trained BERT embeddings [2] into the baseline model and observe enhanced performance in image captioning tasks.
- Finally, we evaluate performances of the two models with quantitative metrics (i.e. cross-entropy loss and BLEU scores). Also, we examine the captions generated by the two models with qualitative focus.

2. Related Work

Several methods have been developed for generating image captions, and most of them have utilized recurrent neural network (RNN) and a sequence-to-sequence encoder-decoder framework, first introduced in neural machine translation tasks [1]. Since then, a new paradigm of image caption has been proposed. First, a new approach of replacing RNN encoder with a deep Convolutional Neural Network (CNN) has been presented in [6]. They confirmed that CNNs are able to produce rich representations of input images by using a fixed length of embeddings. In [6], the authors pre-trained CNNs for an image classification task and passes the CNN's last hidden layer to the RNN decoder that generates captions.

Researchers have also applied the attention mechanism [4] to image captioning tasks. In [8], a soft-attention model consists of an encoder-decoder framework, where its encoder takes a single raw image and feeds it into a CNN to extract a set of feature vectors from the lower convolutional layer. Then, the decoder takes a Long Short-Term Memory (LSTM) network to generate one word at every time step from the encoder's output vectors. Our paper is inspired by this soft-attention model and adopts this CNN-LSTM framework as a baseline model to produce captions.

Due to rapid advancements of language models in NLP tasks inspired by Transformer [4], decoders for text generation tasks have achieved improved performance. For example, BERT [2] learns deep bidirectional representations from unlabeled texts by leveraging a strategy of masked language modeling (MLM). In other words, it learns the left and right context of a word in a sentence. RNN or

LSTM is unidirectional and generates the text that does not fully consider the context of the sentence [1, 8], whereas BERT resolves the problem. BERT has demonstrated its state-of-the-art performance in many NLP tasks, particularly in next-word-prediction tasks. Thus, our works also integrate pre-trained BERT embeddings into the baseline CNN-LSTM model and examine the BERT model's performance in generating captions of certain images.

3. Approach

3.1. Problem Statement

Equation 1 is the mathematical description of the task of image captioning, as proposed in [8]. Taking a single raw image as an input, the two captioning models that we implement are expected to generate a caption $\mathbf{y}$ encoded as a sequence of 1-to-K encoded words, which is a one-hot vector representation of our vocabulary.

$$y = \{\mathbf{y_1}, \mathbf{y_2}, \ldots, \mathbf{y_C}\}, \mathbf{y_i} \in \mathbb{R}^K \tag{1}$$

, where K is the size of vocabulary and C is the length of the caption.

3.2. Models

In order to generate image captions, we implement two attention-based encoder-decoder models, inspired by [8]. Note that there are two variants of attention-based caption generators proposed in [8], but considering the amount of work for the project, we decide to focus solely on the soft-attention variant that are trainable by standard backpropagation methods.

For an encoder, both models use a CNN that takes an input image and produces vector representations of detected objects in the image. Those vectors are then passed to a decoder, a LSTM that attends to the image and produces a descriptive caption of the image one word at one time step.

To summarize, we implement two soft-attention models with the same architecture of CNN-LSTM. The baseline model replicates the soft-attention model proposed in [8]. However, we also extend on this baseline model that integrates pre-trained BERT embeddings to the captions generated by the LSTM decoder, where we expect this variant model to show the advanced performance of the baseline.

3.2.1 Encoder: Convolutional Neural Network

CNN in the encoder receives a raw image as an input, and from its lower convolutional layer of L filters, it produces a feature vector $\mathbf{a}$ that consists of L multiple vectors, where each vector a_i has a D-dimension corresponding to a part of the image [8]. Equation 2 describes the output vector $\mathbf{a}$.

$$\mathbf{a} = \{a_1, a_2, \ldots, a_L\} \in \mathbb{R}^D \tag{2}$$

3.2.2 Decoder 1: Baseline Soft Attention

The baseline model uses a LSTM Network to generate words one step at a time stamp, by conditioning on the previous hidden state h_{t-1}, the context vector z_t, and the previously generated caption words y_{t-1} [8].

$$\begin{pmatrix} i_t \\ f_t \\ o_t \\ g_t \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ tanh \end{pmatrix} T_{D+m+n,n} \begin{pmatrix} Ey_{t-1} \\ h_{t-1} \\ \hat{z_t} \end{pmatrix} \tag{3}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot g_t \tag{4}$$

$$h_t = o_t \odot tanh(c_t) \tag{5}$$

$$\hat{z_t} = \phi(a, \alpha) \tag{6}$$

As observed in Equation 3, 4 and 5, i_t, f_t, o_t and g_t are the input, forget, memory, and output states of the LSTM decoder at the current time stamp t. Also, note that c_t is the cell state at time step t and h_t is the hidden state at t. $\odot$ is an element-wise multiplication and $T_{n,m}$ denotes an affine transformation from dimension n to m.

In Equation 3, Ey_{t-1} represents an embedding vector produced by the embedding matrix E from the previously generated caption word y_{t-1}. $\hat{z_t}$ is a context vector that dynamically represents a relevant part of image at time t. As described in Equation 6, $\hat{z_t}$ is determined by the following three components: (1) CNN output a, (2) the weight vector α that decides which location to "attend" on a, and (3) the soft-attention model f_{att} that computes α. Also, ϕ is the selected mechanism that computes z_t, such as $\sum$. In the soft-attention f_{att}, we use a expected value of z_t over all locations rather than exactly sampling only one location to attend. Also for the soft-attention model, the embedding matrix E is trained by the standard gradient descent.

3.2.3 Decoder 2: BERT Soft Attention

As an enhancement to the baseline model, we implement the same soft-attention CNN-LSTM but with the pre-trained BERT embeddings in its LSTM decoder. Due to the recent breakthrough of BERT, decoders have been able to discern multiple meanings of a word depending on its location in a sentence. In other words, BERT can generate bi-directional contextualized word embeddings conditioned on the context of the word in a sentence [2].

To integrate BERT into the LSTM decoder, we take a batch of captions as our input $\mathbf{c} = \{c_1, c_2, \ldots, c_B\}$, where B is the size of the batch and c_i is the text of a caption. Iteratively taking each word c_i, we perform the following steps for the integration process:

- Tokenize each caption c_i with BERT's WordPiece tokenizer [7]

- Add the special '[CLS]' token to the beginning of each caption c_i
- Pass the tokenized caption into BERT
- Retrieve the output and discard the '[CLS]' token's embedding
- Detokenize the embeddings to retrieve the original text representation of the caption

After performing each step for all caption words in the batch, we obtain caption embeddings $b = \{b_1, b_2, \dots, b_B\}$ where b_i is a tensor of the word's contexualized embedding pretrained on BERT.

4. Experiment

In this section, we explain the detailed descriptions of our experiments, particularly data processing setup, modeling setup, and evaluation metrics setup, respectively. Note that all software implementation procedures are performed in PyTorch framework.

4.1. Data Preprocessing

We used Microsoft Common Objects in Context (MS COCO) dataset [3], publicly released in 2014 for image captioning task. To be specific, we used raw images in the COCO 2014 train/validation datasets as inputs to our models, and the corresponding reference captions were provided in the MS COCO train/Val annotations file. Also, the COCO 2014 training dataset is used for training our models, and the separate COCO 2014 validation dataset is to validate and test the trained models. Note that we did not use the COCO testing dataset for testing the trained models, as the main objective of our paper is to investigate whether a BERT-integrated soft-attention model shows better performance than the baseline soft-attention model in terms of similarity with the reference captions.

The training dataset contains 82,783 raw images and the validation dataset contains 40,504 images. Also, every image has its five reference captions. An example image in COCO training dataset and its corresponding five annotations are presented in Figure 1.

A grey cat sitting by a round mirror.
A grey tiger cat staring at himself in
the mirror. A cat sitting in front of
a mirror near a door. A gray cat looks
at its reflection in a mirror with a
decorative wooden frame. A gray cat
looking at itself in the mirror.

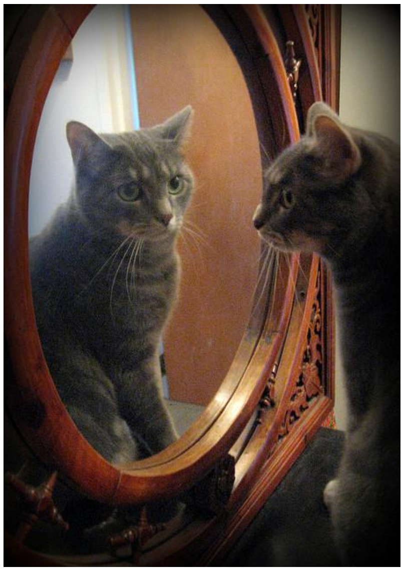

Figure 1. An example image of training set and its corresponding five annotations (above)

We pre-processd the raw MS COCO images and annotations in the training dataset to be adaptable to the objectives of our project. First, we re-size and normalize all the images to 224x224 pixels. Then, we parsed the reference captions, followed by tokenizing them and building a vocabulary with all the 8,853 existing words in the training dataset.

4.2. Learning Process Setup

Our expected result is that the two models encode the preprocessed images from MS COCO and use their decoders to generate captions of the images. The loss function for the optimization process is cross-entropy loss, and we used Adam optimization algorithm as the optimizer of the loss function.

For software implementation of the encoder parts, we used Torchvision's pretrained ResNet-101 as the CNN encoder for the efficiency of our training process. Also, we discarded the last pooling and linear layers of the ResNet-101 because no classification process should be made at this encoder step. Instead, we used an adaptive pooling layer to pass the output vector of fixed size L to the decoder step.

For implementing the baseline soft-attention LSTM decoder, we used the optimized hyperparameters of the soft-attention model suggested in [8], as shown in Table 1. Note that the number of epochs for training is set to 4 due to our GPU usage limits. For implementing soft-attention mechanism in the decoder, we followed an exisitng reference from the Github repository [5].

Lastly, for the BERT soft-attention decoder, we also used the same optimal hyperparameters presented in Table 1. In addition, we used pre-trained BERT base version with 12 encoder layers, 768 hidden units in its feedforward network, and 12 attention heads. To do so, we used the BertModel and BertTokenizer from Huggingface's Transformers [7].

Hyperparameters	Optimized value
Gradient clip	5
Number of epochs	4
Batch size	32
Decoder learning rate	0.0004
Dropout	0.5
Vocabulary size	8,853
Encoder (CNN) dimension	2,048
Attention dimension	512
Initial Weights	Uniform distribution [-0.1, 0.1]

Table 1. Suggested hyperparameters from [8]

4.3. Evaluation Setup

First, we ran the two trained models on the MS COCO validation dataset. Then, we measured the cross-entropy loss and Bilingual Evaluation Understudy Score (BLEU) in order to evaluate the performance of the two models. For further explanation, a BLEU score is one of the popular metrics in the downstream task of text generation in the NLP field because it enables us to compare a machine-generated sentence to a reference sentence. It ranges from 0 to 1, where the score of 1 represents a perfect match and 0 means a perfect mismatch. BLEU-1 and BLEU-4 are cumulative 1-gram and 4-gram BLEU scores, respoectively. We used the NLTK library to compute each BLEU scores.

5. Results

5.1. Baseline Soft Attention

We observed that the trained baseline soft-attention model can generate a caption of a given image from COCO validation dataset. For example, given the following Figure 2, the model successfully generates a caption that is exactly same with the corresponding reference caption but with different order of words, as presented in Table 2.

Type	Caption
Hypothesis	a man holding a tennis racquet and a tennis ball .
Reference	a man holding a tennis ball and a tennis racket .

Table 2. The hypothesis and reference captions of Figure 2

Type	Caption
Hypothesis	a man sitting in a table with a glass wine of front of him .
Reference	a woman standing at a counter with a three bottles in front of her

Table 3. The hypothesis and reference captions of Figure 3

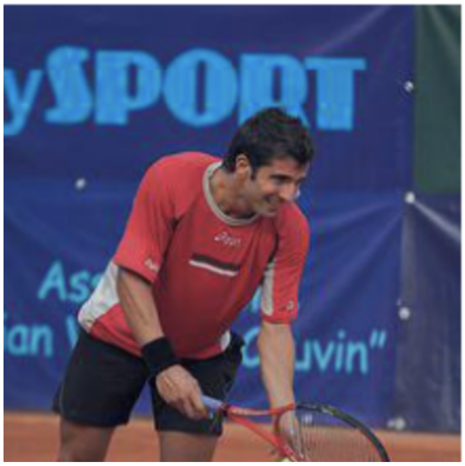

Figure 2. A successful example image of MS COCO Validation

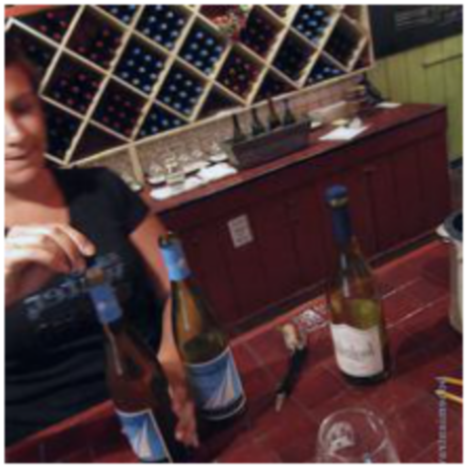

Figure 3. A failed example of MS COCO Validation

However, there are many images of which the baseline model failed to generate precise captions. For example, as observed in Figure 3 and Table 3, the model wrongly recognizes the person in the image as a man and its generated caption also has some grammatical errors. However, we observed that the model is able to capture certain important points of the image, considering that the model predicts similar objects related to ”bottle” in front of the person.

Another failed example from the baseline model is shown in Figure 4 and Table 4. The model generates a grammatically wrong sentence and does not correctly interpret that the person in the image is a young boy. In addition, the words ’a’ and ’cake’ are repeated many times in the generated caption.

We confirmed that the baseline model does not accurately caption images in the COCO validation. This means that the baseline model correctly learned only a few representations of objects in images and understood few text representations of the respective captions, with the hyper-

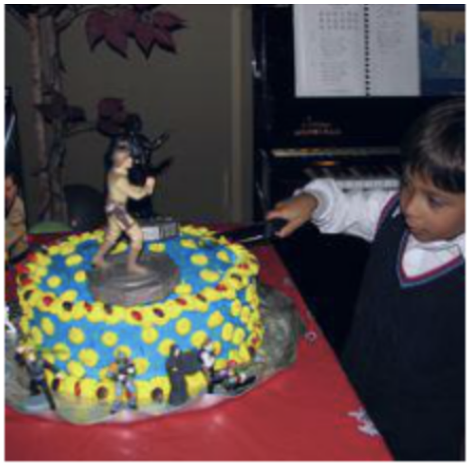

Figure 4. Another failed example of MS COCO Validation

Type	Caption
Hypothesis	a woman boy is a a cake cake cake .
Reference	a young kid cutting into a star wars cake .

Table 4. The hypothesis and reference captions of Figure 4

parameter setup in Table 1. Therefore, we concluded that additional process of hyperparameter tuning are needed for the baseline model to learn all the features in the COCO training dataset.

5.2. BERT Soft Attention

The soft-attention model integrated with the BERT's embeddings generates accurate captions of images in the COCO validation dataset. For example, the BERT soft-attention model produces the exactly same sentence with the reference caption of Figure 5.

Type	Caption
Hypothesis	a cat laying down next to a woman wearing glasses .
Reference	a cat laying down next to a woman wearing glasses .

Table 5. The hypothesis and reference captions of Figure 5

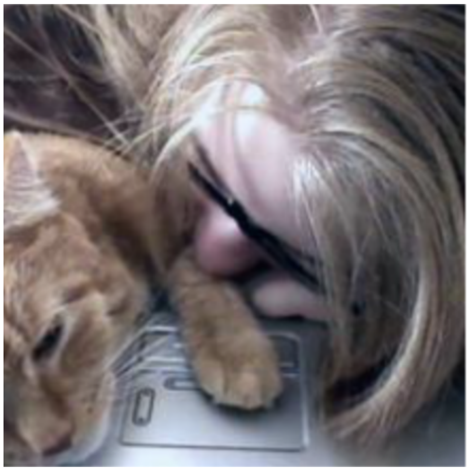

Figure 5. A successful example of MS COCO Validation by the BERT soft-attention model

In addition to accurately generating captions, the BERT soft-attention model is able to understand the scene of images and replace certain words in reference captions with other words that possess similar context. Table 6 demonstrates that the BERT soft-attention model recognizes an elephant and a pole in Figure 6 precisely. Thus, the model predicts similar sentences to the reference caption but translates the following words 'rest' and 'black' into 'stand' and 'wooden', respectively. In addition, the generated hypothesis caption makes sense and does not include grammatical errors. Therefore, we observed the effectiveness of the pre-trained BERT embeddings integrated into the LSTM decoder with soft-attention.

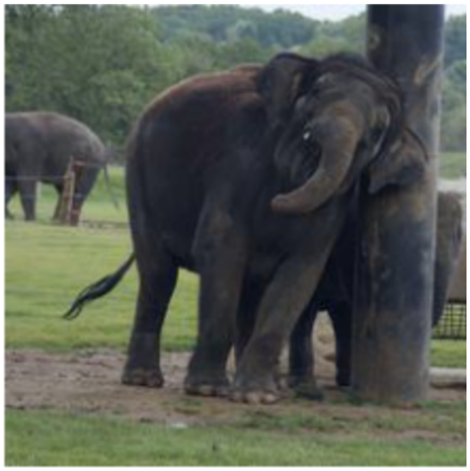

Figure 6. Another successful example of MS COCO Validation from the BERT soft-attention model

Type	Caption
Hypothesis	a large elephant stands on a wooden pole .
Reference	a large elephant rests on a black pole .

Table 6. The hypothesis and reference captions of Figure 6

Although the BERT soft-attention model has shown remarkable performance in generating captions, it also makes sentences that sound unnatural. For instance, the model correctly recognizes a pizza in Figure 7 but repeats the word 'pizza' twice in its generated caption.

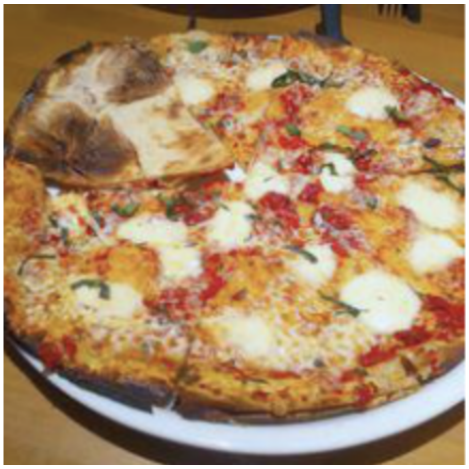

Figure 7. A failed example from the BERT soft-attention model

Type	Caption
Hypothesis	a large pizza pizza sitting on a plate on a table
Reference	a fully baked pizza sitting on a plate on a table

Table 7. The hypothesis and reference captions of Figure 7

5.3. Evaluation of Models

Figure 8 shows the trend of cross-entropy loss when training on the MS COCO training set both the baseline and BERT soft-attention models with the same hyperparameter setup as described in Table 1. With the total of 4 epochs, the BERT soft-attention model decreases its cross-entropy loss significantly, while the baseline model does not reduce the training loss largely. We confirmed that providing the BERT embeddings to the baseline decoder performed better in learning context of words in reference captions.

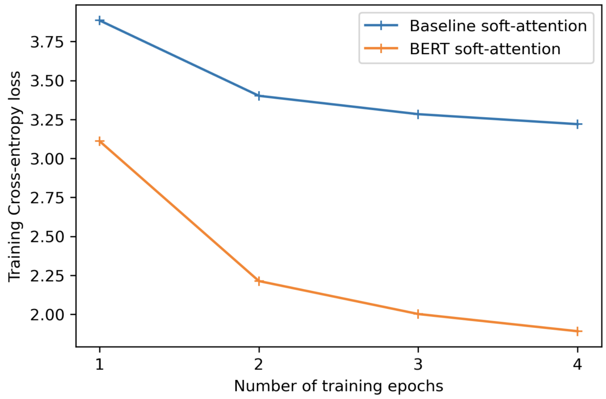

Figure 8. Cross-entropy loss from training process

In addition, we validated the two soft-attention models by running each model with only a single epoch on the COCO validation dataset. In Table 8, the validation loss from the BERT soft-attention model is nearly a half of the loss of the baseline model. This result aligns well with Figure 8 in that the BERT model learned accurate representations of contextualized words in the reference captions of the COCO validation set.

Model	Cross-entropy loss
Baseline	3.021
BERT	1.719

Table 8. Cross-entropy loss on MS COCO validation set (1 epoch)

Furthermore, we computed BLEU scores of the two models by comparing each hypotheses captions to the corresponding 5 reference captions in the COCO validation dataset. Table 9 displays that the BERT soft-attention model has shown significant improvement in all aspects of BLEU scores. Also, all those BLEU scores of the BERT model are within a range from 0.4 to 0.8, which represents that the captions generated by the BERT soft-attention model are identified to be highly fluent. Therefore, integrating pre-trained BERT embeddings into the baseline LSTM decoder generates more accurate and precise captions of images in the COCO validation dataset.

Model	BLEU	BLEU-1	BLEU-4
Baseline	0.146	0.527	0.036
BERT	**0.591**	**0.823**	**0.433**

Table 9. BLEU scores on the COCO validation set

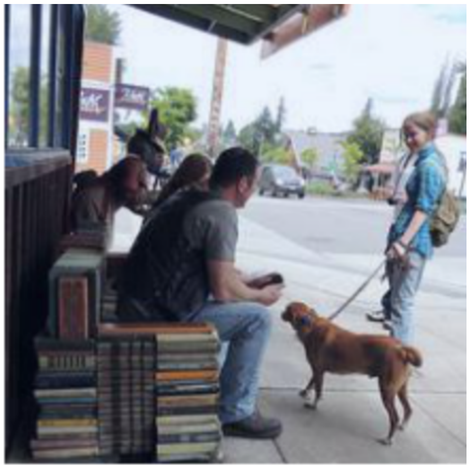

Figure 9. An example image for comparison

Table 10 compares the reference caption of Figure 9 with the hypotheses captions generated by the baseline and BERT soft-attention models, respectively. The baseline model made grammatical mistakes on its caption, whereas the BERT soft-attention model not only captured the context of words in the reference caption but also translated the word ’lady’ into ’woman’ that has similar context. Therefore, we concluded that the BERT soft-attention model outperforms the soft-attention model proposed by [8].

Type	Caption
Reference	a lady with a dog is talking to a lady and man .
Baseline	a group and a dog sitting sitting on a man . a .
BERT	a lady with a dog is talking to a man and woman .

Table 10. The reference caption of Figure 9, followed by the generated captions from the two models

6. Conclusion

In this paper, we implemented two types of soft-attention models: (1) the baseline CNN-LSTM network proposed by [8] and (2) its variant that integrated pre-trained BERT embeddings in the decoder to apply our extended approach to image captioning task. Thus, we confirmed throughout several experiments that integrating pre-trained embeddings into the LSTM decoder significantly enhances performance of the baseline model. To be specific, this approach generates higher quality of captions, under the condition that both baseline and BERT-variant models are trained with the same hyperparameters. In addition, the validation loss is reduced by nearly a half and the BLEU scores are significantly increased, by integrating BERT embeddings into the baseline decoder.

Due to our consideration of simple training process and Google Colab GPU usage limits, the paper implies several limitations. First, we did not train the two soft-attention models until their cross-entropy losses converge to nearly 0. Also, we did not perform hyperparameter tuning: instead, we adopted the suggested hyperparameters from [8]. Thus, possible ideas of future study is to train the two models on the COCO training dataset and find optimized models by performing hyperparameter tuning on the COCO validation dataset. Then, we would finally test our models on the COCO Testing dataset to check the quality of captions generated by those models.

References

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate, 2016. 1, 2

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. 1, 2

[3] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Doll’a r, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014. 3

[4] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017. 1

[5] Sagar Vinodababu. A-pytorch-tutorial-to-image-captioning. 3

[6] Oriol Vinyals, Alexander Toshev, Samy Bengio, and D. Erhan. Show and tell: A neural image caption generator. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3156–3164, 2015. 1

[7] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, Oct. 2020. Association for Computational Linguistics. 2, 3

[8] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron C. Courville, Ruslan Salakhutdinov, Richard S. Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. *CoRR*, abs/1502.03044, 2015. 1, 2, 3, 4, 7