MODEL ARCHITECTURES FOR IMAGE CAPTIONING: CNN-RNN VS CNN-GPT

Authors: Jacob Idoko, Gabriel Gabari, Aakash Sorathiya, Jagrit Acharya, Chioma Ukaegbu, and Emmanuel Alafonye

Department of Electrical and Computer Engineering,
University of Calgary

ABSTRACT

Image captioning refers to the generation of natural language descriptions for visual content. It bridges the gap between computer vision and natural language processing, enabling machines to describe an image.
This work compares two architectures: CNN-RNN, and CNN-GPT. CNN-RNN and CNN-GPT leverage Convolutional Neural Networks (CNNs) for feature extraction from images. Following feature extraction, different approaches for language generation are employed. We evaluate the effectiveness of these models using the Bilingual Evaluation Understudy (BLEU) metric. This analysis aims to identify the model that achieves the best performance in generating accurate and descriptive captions for unseen images in the Flickr8k dataset.

Index Terms— Image Captioning, Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), Generative Pre-trained Transformer(GPT), BLEU metric, Flickr8k dataset

1. INTRODUCTION

Image captioning plays a crucial role in bridging the gap between computer vision and natural language processing (NLP). This technology allows computers to interpret and describe visual content using natural language, similar to how humans do. Image captioning has a wide range of applications, including assisting visually impaired individuals by providing audio descriptions of images, automating image tagging for improved web accessibility, and enhancing image retrieval systems for faster and more relevant search results.

The field has undergone a revolution with the advent of deep learning approaches. These approaches leverage powerful algorithms to automatically learn complex representations of images and generate captions without the need for explicit hand-crafted features[3].

This work compares two deep learning architectures for image captioning, each utilizing an encoder-decoder structure: Convolutional Neural Networks (CNNs) with Recurrent Neural Networks (RNNs) (CNN-RNN), and CNNs with Generative Pre-trained Transformers (GPTs) (CNN-GPT). CNNs excel at extracting informative features from images, while RNNs are adept at handling sequential data like sentences. This synergy has made CNN-RNN models highly successful in image captioning tasks.. Recently, advancements in transformer-based architectures like GPTs, known for their impressive performance in various NLP tasks including text generation, have opened exciting possibilities for image captioning. We aim to identify which architecture achieves the best performance in generating accurate and descriptive captions for unseen images. We will evaluate the models using the popular Flickr8k dataset and the Bilingual Evaluation Understudy (BLEU) metric. An illustration of an encoder-decoder model for image captioning is shown in Fig. 1.

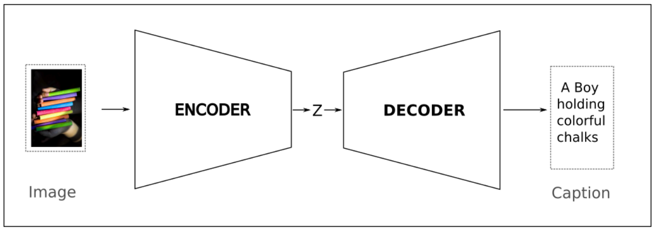

Fig. 1. Visualizing an image captioning architecture

2. RELATED WORK

In 2014, Vinyals et al. pioneered the CNN-RNN architecture, which combines the strength of CNNs in feature extraction with RNNs' sequential processing capability to generate captions[7]. Xu et al. also proposed a model using a CNN for feature extraction and an LSTM (Long Short-Term Memory) network, a type of RNN, for caption generation. Their work achieved significant improvements over traditional methods[5]. Vinyals et al. further advanced the field by introducing attention mechanisms within the RNN decoder, allowing the model to focus on specific image regions relevant to the caption being generated[8].

On the other hand, the CNN-GPT Architecture leverages pre-trained transformer models to directly generate captions from images . They rely on an encoder-decoder architecture similar to CNN-RNNs but utilize self-attention mechanisms to capture long-range dependencies within the data[6].

3. Materials and Methods.

This section aims to provide a comprehensive understanding of the experimental setup and methodology followed in our comparative analysis. We describe the datasets utilized for training and evaluation, the preprocessing techniques to prepare the data, and the model architectures implemented for each approach. Additionally, we outline the training procedures, including optimization techniques, hyperparameter settings, and evaluation metrics used to assess the performance of the models.

3.1 Dataset

We used the Flicker8k dataset in this work. This dataset consists of 8000 images accompanied by five captions, resulting in a total of 40,000 captions for the entire dataset. We split the dataset into training, validation, and testing sets.

The train, validation and testing sets contain 6000, 1000 and 1000 images respectively corresponding to a split of 75%, 12.5% and 12.5% respectively.

3.2 CNN-RNN Model

- **Preprocessing**: Captions are tokenized, excluding words with fewer than five occurrences. Images are resized to 256x256 pixels and normalized. Each caption is adorned with special tokens <start>, <end>, <pad>, and <unk>. Data is stored in HDF5 and JSON files for efficient training processing.

- **Encoder (CNN):** The Encoder utilizes a pre-trained ResNet-101 model. This pre-trained model extracts features from the input images. The final layers of ResNet-101 are fine-tuned during training to adapt to the image captioning task. The encoder outputs a fixed-length feature vector representing the encoded image information.

- **Decoder (LSTM):** The decoder is implemented as an LSTM-based architecture[1]. It takes the image features from the encoder and generates captions. It uses an embedding layer to convert word indices into dense vectors, fed into the LSTM along with the image features. The LSTM generates a sequence of words one at a time, conditioned on the image features and previously generated words. The output of the LSTM is passed through a linear layer followed by a softmax activation to produce a probability distribution over the vocabulary for the next word in the caption.

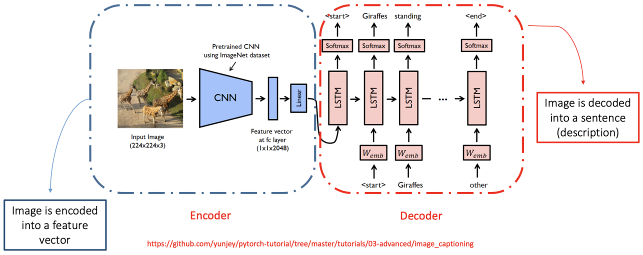
Pretrained CNN using ImageNet dataset
CNN
Input Image (224x224x3)
Feature vector at fc layer (1x1x2048)
Linear
<start>
Giraffes
standing
<end>
Softmax
Softmax
Softmax
Softmax
LSTM
LSTM
LSTM
LSTM
W_{emb}
W_{emb}
W_{emb}
<start>
Giraffes
other
Image is decoded into a sentence (description)
Image is encoded into a feature vector
Encoder
Decoder
https://github.com/yunjey/pytorch-tutorial/tree/master/tutorials/03-advanced/image_captioning

Fig. 2. CNN-RNN Model.

3.3 CNN-GPT Model

- **Preprocessing:** Similar to the preprocessing done in the CNN-RNN model.

- **Encoder (CNN):** The same encoder i.e ResNet-101 used in the CNN-RNN model is used.

- **Decoder (GPT):** We utilize two embedding layers within the decoder. One layer maps words to semantic meaning vectors, while the other assigns positional embeddings to capture word order within the sentence, departing from the original Transformer's fixed positional encoding[6]. We employ PyTorch's nn.TransformerDecoderLayer as building blocks, featuring masked self-attention, cross-attention to attend to encoded image features, and layer normalization for training stability. 6 layers are stacked to create a deeper architecture. 16 attentions heads are used to capture context. During training, the decoder processes the image, a shifted caption (excluding the first word), and a padding mask. Token and positional embeddings are applied to the caption, while the image undergoes feature extraction by a pre-trained CNN(ResNet-101). Cross-attention and masking occur within Transformer blocks, followed by feeding the final output through a linear layer to predict the next word's probability distribution.

The architecture implemented is shown in Fig. 3.

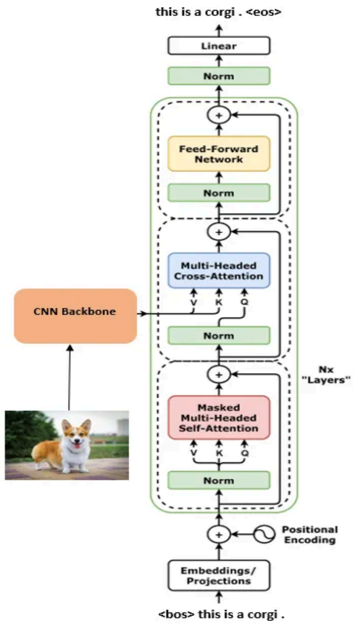

Fig. 3. CNN-GPT Model Architecture.

3.5 Training Details

Both models were trained for 100 epochs on NVIDIA V100 GPU. The CNN-RNN Model took ~5 hours to train while the CNN-GPT took ~11 hours. The parameters used for training are shown in Table 1.

Table 1. Training Parameters

Model	Optimizer	Learning Rate	Batch Size	Epochs	Loss
CNN-RNN	Adam	5e^-4	32	100	Cross Entropy
CNN-GPT	Adam	5e^-4	32	100	Cross Entropy

4. Results and Discussion.

4.1 Training Performance

Both the CNN-RNN and CNN-GPT models were trained for 100 epochs as described in Section 3.5. We analyzed the plots of average training and validation loss (Fig. 5 and Fig. 6) to understand their convergence behaviour. Additionally, we monitored the BLEU-4 score on the validation set after each epoch to track caption generation quality. We selected the model with the highest BLEU-4 score as the best model.

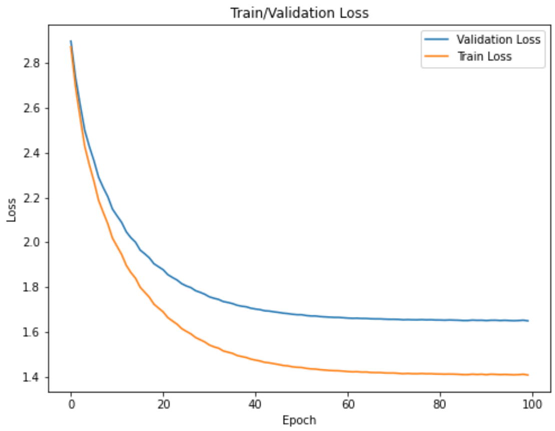

Fig. 4. Average epoch loss for train/validation set (CNN-RNN)

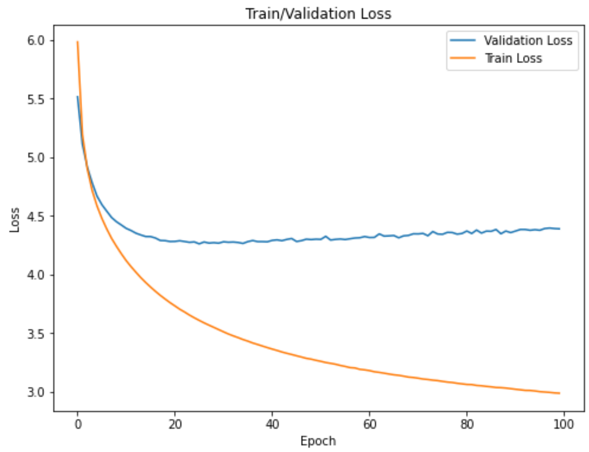

Fig. 5. Average epoch loss for train/validation set (CNN-GPT)

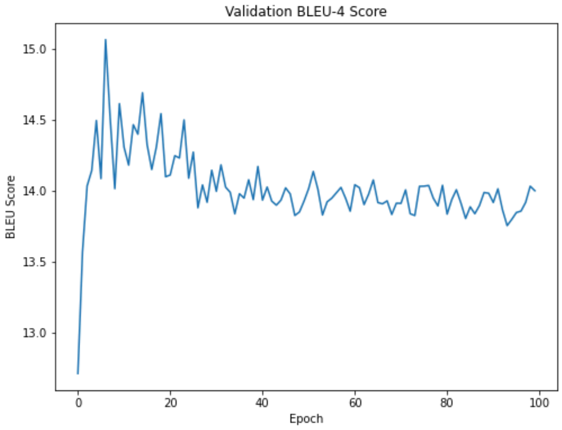

Fig. 6. BLEU-4 scores for validation set (CNN-RNN)

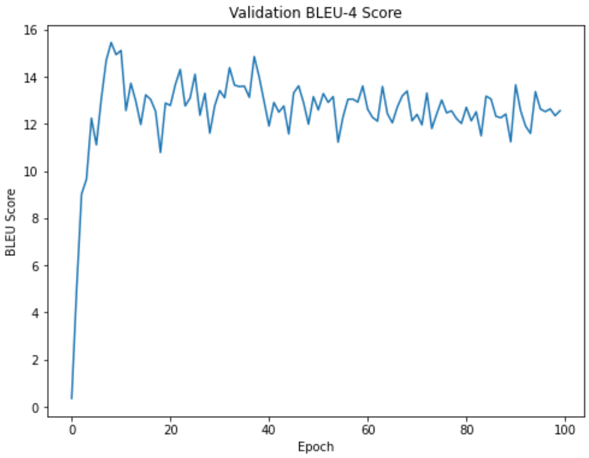

Fig. 7. BLEU-4 scores for validation set (CNN-GPT)

4.2 Evaluation on Test Set

To evaluate the performance of the CNN-RNN and CNN-GPT models on unseen data, we computed BLEU scores on the test set from the Flickr8k dataset. BLEU score is a metric commonly used for image captioning, and it measures the similarity between the generated caption and human-written reference captions for the same image. Higher BLEU scores indicate better caption quality[11].

Table 2 summarizes the BLEU scores achieved by both models on the test set. It reports BLEU scores for different n-gram lengths (1 to 4), where n-gram refers to sequences of n words.

Table 2. BLEU Scores on Flickr8k Test Set

Model	BLEU-1	BLEU-2	BLEU-3	BLEU-4
CNN-RNN	64.25	40.98	24.81	14.63
CNN-GPT	52.65	33.6	20.95	12.27

4.3 Qualitative Analysis

In addition to quantitative evaluation using BLEU scores, a qualitative analysis is crucial for understanding how well the models capture image content and generate coherent sentences. Here, we present captions generated by both CNN-RNN and CNN-GPT models for a few images from the test set. Fig. 8. shows some visualization of images and their generated captions from both models.

5. Conclusion

This work compared the performance of two image captioning architectures, CNN-RNN and CNN-GPT, on the Flickr8k dataset. We evaluated their effectiveness in generating captions for unseen images using BLEU scores. Our analysis revealed that the CNN-RNN model achieved higher BLEU scores across all n-gram lengths (BLEU-1, BLEU-2, BLEU-3, BLEU-4) compared to the CNN-GPT model. This suggests that the CNN-RNN architecture is better suited for generating captions on the Flickr8k dataset that are more similar to human-written references in terms of n-gram overlap.

However, there's room for further exploration on both models:

- Attention Mechanisms in CNN-RNN: While CNN-RNN performed well, it lacks the explicit attention mechanisms present in CNN-GPT. Future work could investigate incorporating attention mechanisms within the CNN-RNN decoder to potentially enhance its ability to focus on relevant image regions when generating captions[8].
- Alternative Architectures: Exploring more advanced architectures like ViT-GPT, which leverages transformers for both image feature extraction and caption generation, could be a promising direction. This architecture might learn richer image representations leading to even better captioning performance[9,10].

In conclusion, this study provided insights into the capabilities of CNN-RNN and CNN-GPT for image captioning. By acknowledging limitations and proposing future directions, this work paves the way for further research in this evolving field.

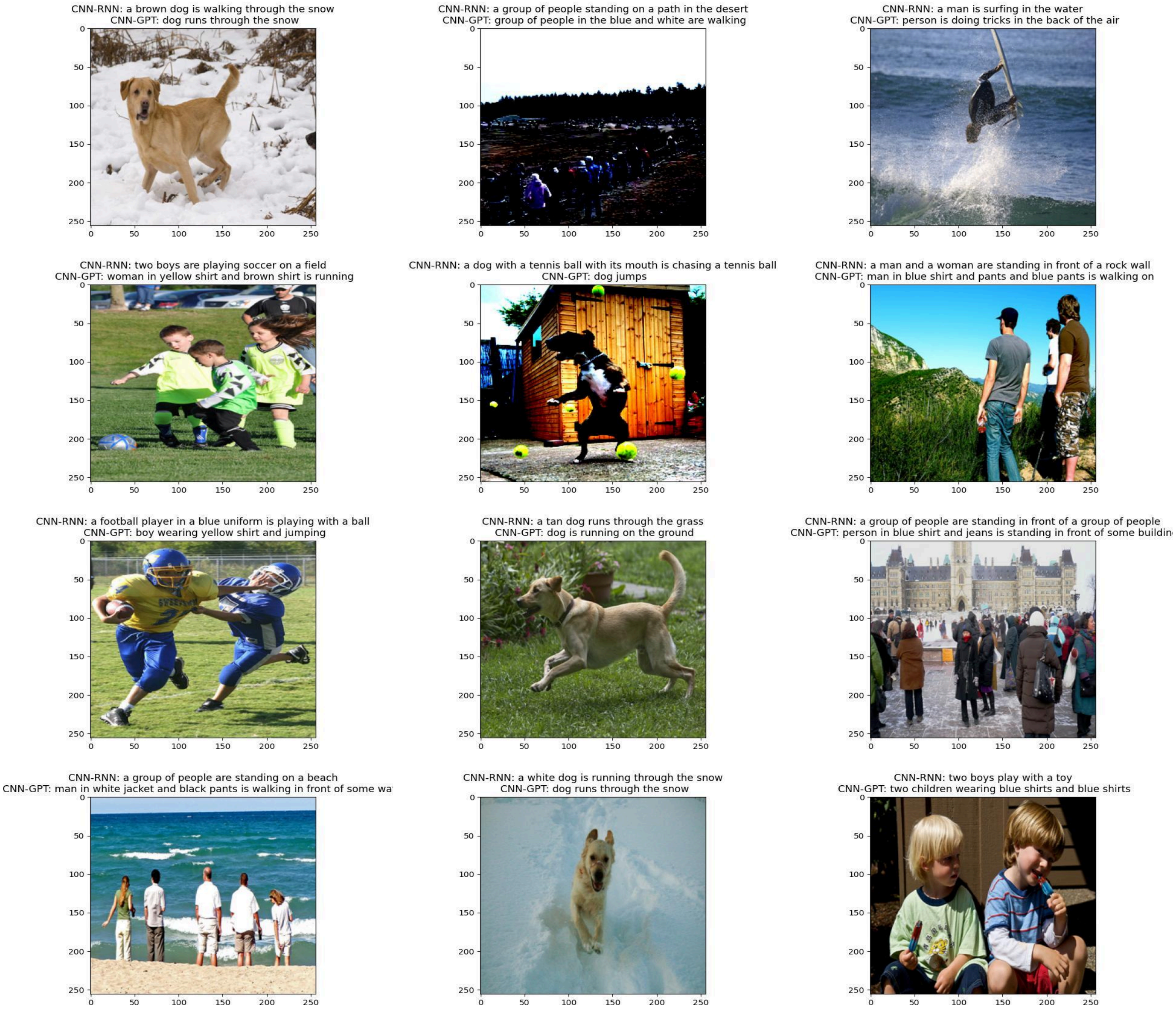
CNN-RNN: a brown dog is walking through the snow
CNN-GPT: dog runs through the snow
CNN-RNN: a group of people standing on a path in the desert
CNN-GPT: group of people in the blue and white are walking
CNN-RNN: a man is surfing in the water
CNN-GPT: person is doing tricks in the back of the air
CNN-RNN: two boys are playing soccer on a field
CNN-GPT: woman in yellow shirt and brown shirt is running
CNN-RNN: a dog with a tennis ball with its mouth is chasing a tennis ball
CNN-GPT: dog jumps
CNN-RNN: a man and a woman are standing in front of a rock wall
CNN-GPT: man in blue shirt and pants and blue pants is walking on
CNN-RNN: a football player in a blue uniform is playing with a ball
CNN-GPT: boy wearing yellow shirt and jumping
CNN-RNN: a tan dog runs through the grass
CNN-GPT: dog is running on the ground
CNN-RNN: a group of people are standing in front of a group of people
CNN-GPT: person in blue shirt and jeans is standing in front of some buildin
CNN-RNN: a group of people are standing on a beach
CNN-GPT: man in white jacket and black pants is walking in front of some wa
CNN-RNN: a white dog is running through the snow
CNN-GPT: dog runs through the snow
CNN-RNN: two boys play with a toy
CNN-GPT: two children wearing blue shirts and blue shirts
0
50
100
150
200
250

Fig. 8. Visualization of images and their generated captions from both models.

References

[1] J. Donahue et al., "Long short-term memory networks for machine translation," in Proceedings of the 34th International Conference on Machine Learning, vol. 70, pp. 1147-1155, 2017. https://arxiv.org/abs/1601.06733

[2] A. Karpathy and L. Fei-Fei, "Deep visual-semantic alignments for generating image descriptions," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3128-3137, 2015. https://arxiv.org/abs/1412.2306

[3] D. Elliott and K. Keller, "Methods for image description," International Journal of Computer Vision, vol. 77, no. 1, pp. 1-43, 2007.

https://support.springer.com/en/support/solutions/articles/6000080471-springerimages

[4] Y. LeCun et al., "Gradient-based learning applied to document recognition," Proceedings of the IEEE, vol. 86, no. 11, pp. 2278-2324, 1998. https://ieeexplore.ieee.org/document/726791

[5] K. Xu et al., "Show and tell: Lessons learned from a caption generation competition," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6106-6114, 2015. https://arxiv.org/abs/1609.06647

[6] A. Vaswani et al., "Attention is all you need," in Advances in Neural Information Processing Systems, vol. 31, pp. 6000-6010, 2017. https://arxiv.org/pdf/1706.03762

[7] Vinyals, O., Toshev, A., Bengio, S., & Erhan, D. (2015). Show and tell: A neural image caption generator. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 3156-3164)

[8] O. Vinyals et al., "Show and attend: Neural image caption generation with visual attention," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2980-2988, 2015. https://arxiv.org/pdf/1502.03044

[9] J. Lu et al., "Hierarchical transformer for image captioning," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 10999-11008, 2019. https://arxiv.org/abs/2207.09666.

[10] Lu, Y., Pu, X., Wang, H., Zhang, Y., & Chen, W. (2020). GPT-2 for image captioning: A better baseline?. arXiv preprint arXiv:2003.05040.

[11] Papineni, K., Roukos, S., Ward, T., & Zhu, W. J. (2002). BLEU: a method for automatic evaluation of machine translation. [Proceedings of the 40th annual meeting on association for computational linguistics (ACL '02), Philadelphia, Pennsylvania, July 7-12, 2002], pp. 311-318. Association for Computational Linguistics.

[12] Gehring, J., Auli, M., Grangier, D., Yarats, D., & Dauphin, Y. N. (2017). Convolutional sequence to sequence learning. In Proceedings of the 34th International Conference on Machine Learning, vol. 70, pp. 1243-1252. https://arxiv.org/abs/1705.03122