Incorporating Attention Mechanism and Word Embeddings for Generating Image Captions

Desiree Mary Dmello **Garima Bajwa**
Department of Computer Science, Lakehead University, Thunder Bay, Ontario, Canada
{dmellod, garima.bajwa}@lakeheadu.ca

Abstract

The main goal of image captioning, a combination of computer vision and NLP, is to provide interpretations of the image in the form of meaningful captions in an automated manner without human intervention. This work provides insight into utilizing a soft-attention mechanism that enables a model to understand how to generate descriptive captions automatically. We considered two approaches - word embeddings trained from scratch and pre-trained GloVe word embeddings to understand if pre-trained vector representations help achieve more meaningful and correct caption expressions than vector representations trained from scratch. This study used visualization to demonstrate how the attention model could concentrate on critical elements of the image while producing the words that corresponded in the output sequence. The research visually represents the captions created by the word embeddings trained from scratch and the pre-trained GloVe embeddings. Evaluation using standard BLEU metrics has demonstrated that our technique significantly enhances model performance.

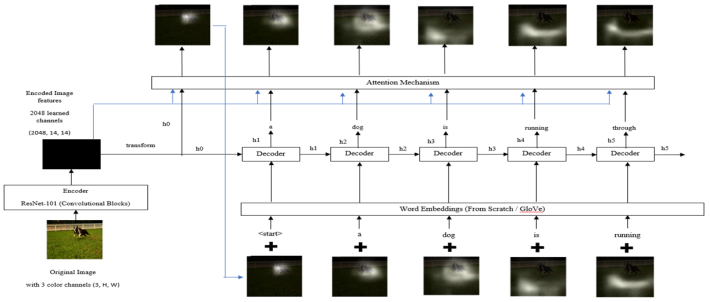


Fig. 1: The Encoder-Decoder Architecture for the Image Captioning Model with the Attention Mechanism. The model displays how the image is first encoded into feature vectors and then passed through the decoder wherein the weighted average across all pixels is computed by the attention model and this weight along with the previous hidden states are fed into the decoder for interpreting the next word for generating the caption.

1 Introduction

Automatically creating captions for images is crucial to scene interpretation, one of computer vision's main objectives [1–3]. Creating image captions can help visually impaired users and make it simple for people to browse and organize vast volumes of usually unstructured visual data. In addition to being able to identify the objects in an image using computer vision, caption generation models also need to capture those objects' connections in natural language. Therefore, caption generation has typically been seen as a challenging subject. This is a significant challenge for machine learning and AI research since it equates to imitating the exceptional human capacity to condense enormous quantities of crucial visual information into descriptive language.

Most image captioning models [4, 5] in addition to an attention mechanism, utilize an RNN [6] or LSTM for the decoder to generate a sequence of words for creating the captions. These architectures use the input captions to generate the word sequences, sending the words to the output not as a string of words but as a vector of numbers, wherein each word is given an index number before the word embeddings are created. These indexed words may be trained from scratch to obtain word embeddings using either representation that has already been assigned to each word or representations that have been customized for the model during training. Finding a relationship between word vectors and numerical vectors may be done in various ways. One-hot encoding is one of the simplest methods, whereas GloVe [7] is a more sophisticated method.

Motivated by recent developments in caption generation and inspired by recent successes in applying attention to object recognition [8, 9] and machine translation [10], this paper focuses on models that can pay attention to the salient part of an image while creating its caption. The following are the contributions of this paper:

- Introduce an attention mechanism that uses a SoftMax function to distribute attention among various parts of an image, facilitating the creation of more descriptive captions.
- By visualizing "where" and "what" the attention was focused on, the paper demonstrates how to acquire insight and interpret the findings from this approach.
- Providing insight through visualization on the usefulness of generating captions through utilizing both word embeddings from scratch and pre-trained GloVe [7] embeddings.
- Quantitatively validate the usefulness of attention in caption generation on the Flickr30k dataset using the BLEU [11] metrics for evaluation.

2 Methodology

This section details the model's architecture, which comprises an Encoder, Decoder, and Attention Network. It also provides details about the Soft Attention and GloVe Embedding Layer utilized by the model (Fig. 1).

2.1 Encoder: CNN (Convolutional Neural Network)

The model takes the input as a raw image and generates a caption 'r' based on a sequence of '1 of K' encoded words.

$$r = \{r_1, ..., r_L\}, r_i \in R^K \tag{1}$$

where L is the caption length and K is the vocabulary size.

A Convolutional Neural Network (CNN) is utilized to extract the feature vectors from the image. In particular, 2048 x 14 x 14-dimensional feature outputs of ResNet-101's [12] final convolutional layer were utilized, allowing the decoder to focus on certain regions of the image by weighing a portion of all feature vectors [8].

2.2 Decoder: Long Short-Term Memory (LSTM) Network

For the decoder, since a sequence needs to be generated for the captions, a Recurrent Neural Network (RNN) is considered. Here, the LSTM cell [13] is used, which generates one word at each time step conditioned on a context vector, the prior hidden state, and the previously generated words obtained through the word embeddings (word embeddings from scratch or GloVe embeddings[7]) for creating the caption. The LSTM cell outputs for the hidden state 'h' at time step 't' and the cell state 'c' at time step 't' would be,

$$h_t, c_t = f_{LSTMCell}(E_{y_{t-1}}, g_t, h_{t-1}, c_{t-1}) \tag{2}$$

where $E_{y_{t-1}}$ is the sequence of word embeddings in the output at the $t-1$ time step.

The relevant portion of the input image is continuously represented by the context vector. The context vector is generated from

Table 1: On the Flickr30k datasets, we compared the performance of the BLEU metrics results on our model (with word embeddings trained from scratch) termed as 'Our Method' and our model (with pre-trained GloVe embeddings) termed as 'Our Model' (GloVe) on the Karpathy Splits [2] and 80/10/10 splits on beam size 1. The measures that do not have scores in the referenced papers have been left empty.

Karpathy Split				
Models	**BLEU-1**	**BLEU-2**	**BLEU-3**	**BLEU-4**
Karpathy et al.[2]	0.573	0.369	0.240	0.157
Mao et al.[3]	0.5479	0.2392	0.1952	-
Vinyals et al.[11]	0.663	0.423	0.277	0.183
Soft-Attention[8]	0.667	0.434	0.288	0.191
Our Method	0.6070	**0.4252**	**0.2942**	**0.2054**
Our Model (GloVe)	0.6210	**0.4366**	**0.3015**	**0.1920**
80/10/10/ Splits				
Our Method	0.6131	**0.4175**	**0.2794**	**0.1883**
Our Model (GloVe)	0.6295	**0.4295**	**0.2866**	**0.1920**

the features extracted by the feature vectors at different image locations. To calculate the context vector for each part of the image, the decoder uses the feature vectors of the last convolutional layer of the CNN. An attention network is used to compute the weighted average across all the features in the image to determine the weight of each of the feature vectors.

At each stage, the next word is created by concatenating this weighted representation of the image with the word from the word embeddings that were previously generated. A mean average of the feature vectors fed through two different single-layer feedforward networks predicts the initial memory state and hidden state of the LSTM.

2.3 Word Embeddings

We employ pre-trained word embeddings and those trained from scratch to process the input word sequence. Obtained an embedding vector of length m for each word, where m in the instance of pre-trained GloVe embeddings equals 300 and for word embeddings from scratch equals 512. The LSTM receives a '$n \times m$' matrix as input, where n is the sequence length, and m is the number of features in the input for a sequence of n words. It can be denoted as,

$$X = x_1, x_2, ..., x_n \tag{3}$$

where n is the sequence's length produced by the embedding layer.

$$E = e_1, e_2, ..., e_n \tag{4}$$

where n is the sequence's length and $e_i \in R^{m \times K}$ with the embedding dimension as m and the vocabulary size as K.

2.4 Attention Network

Along with the word embeddings from the input sequence, the attention network also includes the context vector 'c_t' which is calculated using the attention network as follows:

$$c_t = A(V, h_t) \tag{5}$$

where $V = [v_1, \ldots, v_k], v_i \in Rd$ is the feature vectors, each of which represents a part of the image in d-dimensional space, 'h_t' represents the hidden state of the RNN at time step 't' and 'A' is the attention function [14].

A single layer neural network was used, followed by a SoftMax function to produce the attention distribution in the 'k' areas of the image given the image feature $V \in R^{d \times k}$ and the hidden state $h_t \in R^d$ of the LSTM:

$$g_t = p_h relu(P_v V + (P_g h_t)1) \tag{6}$$

$$\alpha_t = softmax(g_t) \tag{7}$$

where $P_v, P_g \in P^{k \times d}$, and $p_h \in R^k$ are parameters to be learned. $1 \in R^k$ is a vector to set all elements to 1 and $\alpha \in R^k$ is the attention weight over features in 'V'. Based on the attention distribution, once the weights sum up to 1, the context vector 'c_t' can be obtained by:

$$c_t = \sum \alpha_i v_i \tag{8}$$

where α_i and v_i are a set of feature vectors with their corresponding weights.

To encourage the attention model to pay equal attention to every part of the image during training, a doubly stochastic regularization is used. Here, soft attention is considered [8] wherein the attention weights of the feature vectors 'v' in the image sum up to 1 at each time step 't'.

$$\sum \alpha_{vt} = 1 \tag{9}$$

In this case, however, the attention $\sum \alpha_{vt} = 1$ does not contain any constraints. This results in the decoder not attending to certain parts of the input image. To resolve this a penalty is introduced to the attention as:

$$\sum \alpha_{vt} \approx 1 \tag{10}$$

where attention weights in a single feature vector 'v' are encouraged to add up to 1 in all timesteps 'T'. By doing this, the model will be able to pay attention to every feature while generating the entire sequence for the caption. This helps to improve performance and allows the model to generate more descriptive captions [8].

The soft attention model further estimates a gating scalar 'β' from the prior hidden state 'h_{t-1}' at each time step 't' such that,

$$\beta_t = \sigma(f_\beta(h_{t-1})) \tag{11}$$

where $f_\beta(h_{t-1})$ is a linear transform with sigmoid activation of the prior hidden state of the Decoder. This gate's application enables the attention network to focus more attention on the image's objects. By reducing the penalty difference between 1 and the total weight of each feature across all timestamps, the soft attention network is trained.

2.5 Data, Training, and Validation

The experiments were carried out using the Flickr30k dataset [15]. The 31,783 images in the Flickr30k dataset were taken directly from six different Flickr groups. Andrej Karpathy's [2] training, test, and validation split consisting of 29,000 images for the training, 1014 images for validation and 1000 images for testing sets each were utilized. In addition to the Karpathy's [2] split, we also considered a split of 80/10/10 for training, validation and testing of the Flickr30k dataset [15].

A single layer LSTM with a hidden state of 512 was used for the experiments. For the language model and CNN, an Adam optimizer was considered with base learning rates of 4e-4 and 1e-4, respectively. The batch size was set to 80 for scratch-trained word embeddings and 64 for GloVe word embeddings [7]. The training was carried out for up to 20 epochs. If the validation BLEU scores [16] did not improve in the last 8 epochs, early stopping was performed. No fine-tuning was performed on the encoder or the pre-trained word

Fig. 2: Displays the visualization of attention at each time step for samples from the Flickr30k test data. The image is sampled from the Flickr30k dataset using word embeddings from scratch for generating the caption with a beam size of 1.

embeddings.

3 Results

3.1 Quantitative Results

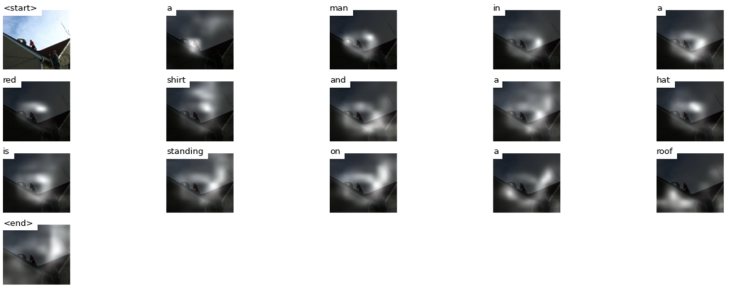
The methods in Table 1 were chosen to compare our method on the basis of their similarities with our approach, as well as the enhancements the researchers suggested in the current procedures. Karpathy et al. [2] developed a multimodal recurrent neural network (m-RNN) architecture using a bidirectional RNN with a hidden layer size of 512 neurons as the decoder and a region convolutional neural network (RCNN) as the encoder. To extract image features for their m-RNN model, Mao et al.[3] employ the AlexNet CNN and two levels of word embedding. To create the caption, the outputs from the second word embedding layer, RNN, and CNN are combined and sent into the 512-dimensional multimodal layer.

In their proposed encoder-decoder system, Vinyals et al.[11] used Inception CNN as the encoder and an LSTM with 512 neurons in the hidden layer as the decoder. According to Xu et al.[8] for their attention-based model, the decoder concentrates on certain areas of the image at each time step to provide captions that are more relevant to the image. In comparison to all these methods, our method outperforms all the other model's methods for BLEU [16] scores BLEU-2, BLEU-3, and BLEU-4 with the exception of BLEU-1 using the word embedding trained from scratch as well as the pre-trained GloVe word embeddings for beam size of 1, 3 and 5 respectively. The beam size = 1 provides a fair comparison to the other models as they did not implement a beam search on their results.

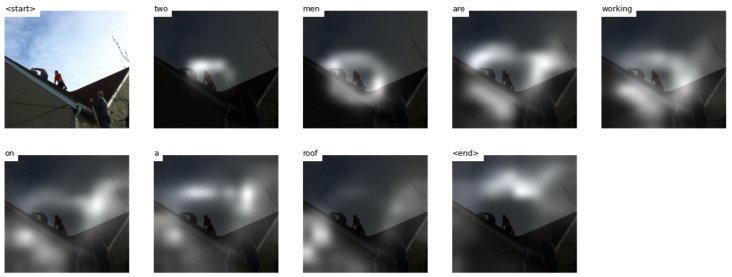
3.2 Qualitative Results

1. Can we leverage attention models to extract main objects from the images for image captioning?

   Fig. 2 provides some examples of the visualization of the captions generated by the attention model with word embeddings trained from scratch for a beam search size of 1. The white blurred parts of the image indicate where the attention weights were focused to identify the next object in the image that needed to be predicted by the model. The model also outperforms other methods in the BLEU scores in Table 1, indicating that attention enables the model to focus on important objects in the image based on the image features it is trained on along with the word embeddings passed in the decoder with the previous hidden state, enabling the model to generate better descriptive captions. In Fig. 2 it can be seen that the model correctly identified a group of people on a snowy path. This example proves that adding an attention mechanism into the model does help the Encoder and Decoder model to identify correct objects from the image when utilizing image features obtained from ResNet-101 Convolutional Neural Network.



(a)



(b)

Fig. 3: Displays the visualization of the captions generated for samples from the Flickr30k test data. Fig. (a) Displays the captions generated by the model where the word embeddings were trained from scratch for a beam size of 1 and Fig. (b) Displays the captions generated by the model for the pre-trained GloVe embeddings with a beam size of 1.

2. Can pre-trained word embeddings like GloVe enhance the quality of captions generated by the image captioning model?

   From Fig. 3(b) it can be seen that the captions generated by the model using the GloVe embeddings [7] were able to correctly identify two men working on the roof and provided a more descriptive and short caption in comparison to Fig. 3(a) wherein the captions generated by the model from training the word embeddings from scratch generated a description which is longer and not exactly necessary for just describing the main details in the image. From Table 1 it can also be seen that word vectors obtained from pre-trained GloVe embeddings performed better than the word embeddings trained from scratch based on the BLEU [16] metric. These examples show that since GloVe is trained on a large language corpus, it is able to provide more meaningful captions. This indicates that utilizing pre-trained word embeddings like GloVe can help image captioning models provide more semantically correct captions with greater ability to generalizations when utilized with an attention mechanism.

4 Conclusion

This research demonstrates how the learned alignments closely resemble human intuition and how this learned attention mechanism may be used to increase the interpretability and generalizability of the model generation process. Using BLEU measures, the attention-based model with and without pre-trained word embeddings outperformed other prior models in the evaluation of the BLEU-2, BLEU-3 and BLEU-4 scores; however, additional fine-tuning incorporated into the encoder model could help further improve the results of the attention model. Even though the attention model with and without pre-trained word embeddings like GloVe was tested on image captioning, it could still be applied to various other domains in NLP and Computer Vision.

References

[1] H. Fang, S. Gupta, F. Iandola, R. K. Srivastava, L. Deng, P. Dollár, J. Gao, X. He, M. Mitchell, J. C. Platt *et al.*, "From captions to visual concepts and back," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1473–1482.

[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*, 2015, pp. 3128–3137.

[3] 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.

[4] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang, "Bottom-up and top-down attention for image captioning and visual question answering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 6077–6086.

[5] S. J. Rennie, E. Marcheret, Y. Mroueh, J. Ross, and V. Goel, "Self-critical sequence training for image captioning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7008–7024.

[6] T. Mikolov, M. Karafiát, L. Burget, J. Cernockỳ, and S. Khudanpur, "Recurrent neural network based language model." in *Interspeech*, vol. 2, no. 3. Makuhari, 2010, pp. 1045–1048.

[7] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.

[8] 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*. PMLR, 2015, pp. 2048–2057.

[9] J. Ba, V. Mnih, and K. Kavukcuoglu, "Multiple object recognition with visual attention," *arXiv preprint arXiv:1412.7755*, 2014.

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

[11] 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*, 2015, pp. 3156–3164.

[12] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[13] R. Kiros, R. Salakhutdinov, and R. Zemel, "Multimodal neural language models," in *International conference on machine learning*. PMLR, 2014, pp. 595–603.

[14] J. Lu, C. Xiong, D. Parikh, and R. Socher, "Knowing when to look: Adaptive attention via a visual sentinel for image captioning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 375–383.

[15] Hsankesara, "Flickr image dataset," Jun 2018. [Online]. Available: https://www.kaggle.com/datasets/hsankesara/flickr-image-dataset

[16] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, 2002, pp. 311–318.