An Approach to Image Captioning with CNN and GRU

Chandramani

University of Maryland College Park.

Abstract

In this paper, I propose a novel approach to image captioning that uses a combination of a convolutional neural network (CNN) and a recurrent neural network (GRU). The CNN is used to extract features from the image, while the GRU is used to generate the caption. The model is trained on the MS-COCO 2017 dataset, which contains over 183k images and captions. I present three different implementations of the model. The first implementation is a baseline model that learns its own embeddings for the vocabulary along with a basic RNN. The second implementation uses the pre-trained GloVe word vectors, which are a set of word embeddings that have been trained on a large corpus of text. The third implementation uses the GloVe word vectors as well as using Gated Recurrent Units to solve the vanishing gradient problem in the network. The results show that the third implementation, which uses the GloVe word vectors and Gated Recurrent Units, achieves the best performance. This suggests that using pre-trained word vectors and Gated Recurrent Units can improve the performance of image captioning models. The model is evaluated using the perplexity and BLEU score metrics. The perplexity metric measures the average number of words in a sentence that are incorrect, while the BLEU score metric measures the similarity between a generated sentence and a reference sentence. The results of this paper suggest that the proposed approach to image captioning is a promising approach that can be used to generate high-quality image captions.

Keywords: Natural Language Processing, Recurrent Neural Network, Convolution Neural Network, Image Captioning, Gated Recurrent Unit

1 Introduction

The ability to automatically obtain a description for a picture has become increasingly important in various fields. For instance, autonomous robotic systems require

accurate image recognition to recognize and process what they see. Another application is creating a searchable image database, without the need for manual tagging and description of images. These applications have become more relevant as technology advances, making the task of determining sentence-level descriptions of images crucial.

Over the past several years, top researchers in the field have attempted to solve this problem, implementing models that can produce natural-sounding sentences that accurately describe the content of images. Major companies, such as Microsoft and Google, have also published models that attempt to solve the problem.

In this paper, I present a model that is inspired by previous researchers' work on the problem. I aim to compare my model's results with the state-of-the-art results in the field. The model is built using deep learning techniques such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), specifically a GRU attention model. I use the image feature vectors extracted by a pre-trained CNN, such as Inception, to capture the visual information of the image. The RNN component generates a sentence by predicting each word in the caption based on the previous words. The attention mechanism helps the model focus on relevant parts of the image while generating the caption.

By training the model on a large dataset of image-caption pairs, I aim to achieve state-of-the-art performance in generating coherent and natural-sounding descriptions of images. This has various potential applications, such as assisting visually impaired individuals in understanding images or improving image search algorithms.
While some approaches to image captioning involve finding the best caption in a pre-existing database that matches the image, this method has limitations. While it ensures that the resulting captions are naturally written and easy to understand, it can fail to describe images that feature unique combinations of objects or objects presented in an unusual way.

An alternative approach is to generate novel image captions using a combination of image features and a language model. This allows for more creative and varied captions that are not limited by the existing database. Generating novel image captions is a challenging and highly useful task that has attracted the attention of researchers and companies alike, as it has many practical applications, such as assisting visually impaired individuals in understanding images and enabling keyword-based image search.

2 Related Work

Cheng Wang et al. [1] proposed a deep bidirectional end-to-end trainable LSTM (Long-Short Term Memory) model for image captioning. The model consists of a deep convolutional neural network (CNN) and two distinct LSTM networks. CNN is used to extract features from the image, while LSTM networks are used to generate the caption. The model is able to learn long-term visual-language interactions by considering both high-level semantic space and recent and historical context knowledge. To avoid overfitting, the authors developed data augmentation strategies such as multi-crop, multi-scale, and vertical mirrors. They also visualized the evolution of bidirectional LSTM internal states over time to see how their models "translate" images to language.

Girish Kulkarni et al. [2] proposed a two-part method for automatically generating natural language descriptions from images. The first phase, content planning, smooths the output of computer vision-based detection and identification algorithms with data acquired from enormous volumes of visually descriptive text. The second phase, surface realization, selects words to generate natural language sentences based on expected content and general statistics from natural language. The authors presented a variety of surface realization methods and compared them using automatic evaluations of descriptions from the proposed generating system against descriptions from other systems. Mark Yatskar et al. [3] proposed using densely labelled images to generate descriptive statements. They gathered human annotations on images of things, parts, qualities, and activities. They used rich annotations of photos to examine description generation. They were able to use the annotations to not only construct new models but also to study what visual information is important for description generation and to generate human-like sentences. Their studies highlighted the importance of activity and bounding-box information, which can be used in research topics. Object grouping and referencing are crucial in more complicated images, such as those with several sentient things, in order to provide acceptable descriptions. Annotations of increasing complexity can be used to investigate issues of those nature.

Yufan Zhou [4] et al. have proposed a way to train text-to-image generation models without using text data by producing text features from image features using the pre-trained CLIP model, achieving state-of-the-art results in conventional text-to-image generation challenges. Their method also reduces training time and expenses for text-to-image creation models. Zihang Meng [5] et al. have presented a method to retrieve related objects from the training set for better object coverage in transformer-based models, leading to significantly improved results in supervised settings. Their approach works well even with non-English image captioning with fewer annotations. Both approaches are supported by strong empirical evidence.

3 Dataset

The Microsoft COCO 2017 dataset [6]is a large-scale dataset of images and annotations. It contains 163,833 images, each of which is associated with a set of annotations. The annotations include a sentence-level description of the image, as well as a list of the objects that are present in the image. The objects are categorized into 80 object classes and 91 stuff classes.

The images in the COCO dataset are divided into three sets: training, validation, and testing. The training set contains 118,000 images, the validation set contains 5,000 images, and the test set contains 41,000 images. The training set is used to train the model, the validation set is used to evaluate the model's performance, and the test set is used to measure the model's final performance.

The COCO dataset is a valuable resource for researchers working on object detection, segmentation, and captioning. It is a large and diverse dataset, and it provides a wide range of annotations that can be used to train and evaluate models.

4 Approach

4.1 Preprocessing of the Data

In image captioning, it is important to have proper data preparation before training the model. In this context, the following steps have been performed:

First, two files have been created - "val_img_ids" and "test_img_ids" - which contain the image ids for all validated and test images, respectively.

Next, the captions have been pre-processed in several ways. All empty spaces, alphanumeric characters have been removed from the captions, and the captions have been converted to lowercase. After that, each caption has been converted into a vector of space-separated words.

To create a vocabulary, all the words appearing more than 5 times in the training dataset have been included. The train captions have been tokenized, and each token has been mapped to its length. These steps are essential to creating a well-organized dataset for training the image captioning model.

4.2 Baseline Model

The chosen model for the project is a multimodal neural network that comprises a convolutional neural network (CNN) and a recurrent neural network (RNN). The CNN is utilized to extract image features while the RNN serves as a language model to predict the next word based on the previous words and the image features provided by the CNN. Although both CNN and RNN are important for the project, the focus is on RNN as it is more relevant to natural language processing.

Instead of training a CNN from scratch, a pre-trained Inception v3 CNN model is used to extract image features. Inception v3 is a state-of-the-art CNN implementation used to predict the probabilities of whether a given image contains classes of images. The model has 1000 classes of objects, and running it on an image returns a 1000-sized vector, which represents the probability of each class. By using a pre-trained CNN, the project saves the time and resources that would have been required to train the CNN from scratch. The extracted features from CNN are then passed through a recurrent neural network (RNN) to generate the caption. The RNN is a type of neural network that can process sequences of inputs and use its internal state to remember information about the previous inputs.

The RNN used in image captioning is a single-layer network with a hidden state of length 512. This hidden state acts as a memory for the network and is updated at each time step. The length of the hidden state was experimentally determined to be the optimal size for generating accurate captions. At each time step, the RNN takes the previous hidden state, the word vector for the current input word, and the CNN output for the image being described, as inputs. The word vector for the input word is obtained by mapping the word to a high-dimensional vector using an embedding layer. This combination of inputs is used to compute the next hidden state and generate the output word.

By incorporating information from both the image and the previous words in the caption, the RNN is able to generate a coherent and relevant caption that accurately

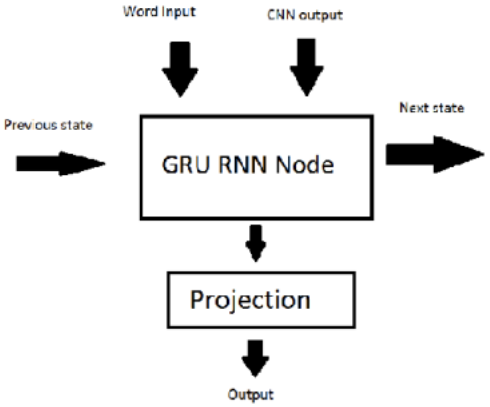

Fig. 1: One Node in GRU (RNN) Model

describes the content of the image. The equation for the next hidden state is equal to:

$$State_i = \sigma(State_{i-1} * H + Word_i * I + Image * N) + b \tag{1}$$

Where H is (Hidden size X Hidden size), State is the hidden state of the RNN, I is (Word Embedding Size X Hidden Size), Word is the word vector obtained by looking up the current input word in embedding, N is (Size of CNN output X Hidden Size), and Image is the output from the CNN and b is the bias. A drawing of one node in the RNN is found in Figure 1. σ) is sigmoid function which is used as the non-linearity of the RNN. The output of each time step is then projected to predict the probability of each word appearing next in the sentence description of the word. The final projection into the vocab size is computed in a multimodal layer that adds a weighted output of the projections of the RNN and the CNN. The equation for the projection at each time step is equal to:

$$output = A * (RNN_i * U) + (1 - A) * (CNN * V) + b \tag{2}$$

Where A is a variable learned by the network that ensures calculates how much of the RNN or CNN output should be included in the final projection. RNN_i) is the output of the RNN at time step i, U is (Hidden Size X Vocab Size), CNN is the output of the CNN for the given image, V is (Size of CNN output X Vocab Size) and b is the bias. This weighted sum of RNN and CNN output ensures that both the image features and the language model both factors in the final output probability at each time step and the best word for both inputs is most likely to be selected.

.

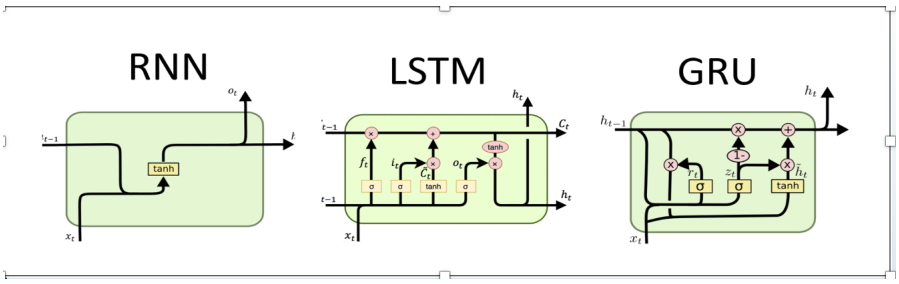

Fig. 2: RNN vs LSTM vs GRU

4.3 GloVe Vector Based Model

The previous model did not generate particularly good results. While its captions might capture some of the semantic meaning of the image, the language model was relatively poor. I reasoned that this was because although I had lots of examples in my dataset, most captions were relatively short. This meant that there was not enough text to adequately train the word vectors that were being used in the embedding of the vocabulary.

To address this issue, I decided to use the pre-trained GloVe word vector. GloVe word vectors are word vectors that are trained based on the co-occurrences of word pairs in a very large corpus. Since the corpus used to train GloVe vectors was substantially larger than the corpus of sentences used in the image captions I had, using GloVe vectors would result in a better representation of the word similarities and would create a better language model.GloVe word vectors are trained using a statistical method called co-occurrence matrix factorization. Co-occurrence matrix factorization is a method of finding patterns in the co-occurrence of words in a corpus. Another reason I thought it may be because of the vanishing Gradient problem. The value of the gradients diminishes rapidly as they are back-propagated to previous states and they have smaller and smaller changes on the variables in the states until the change is insignificant and the variables at states a few steps away do not train at all

4.4 Gated Recurrent Unit Model

GRUs are a more complicated node structure than a base RNN in which the value of the previous state is scored for how important it is in the next state and only as much as is optimal is used to compute the next state. This helps alleviate the vanishing gradient problem as the values of the gradients are no longer exponentiated at each step in backpropagation and the gradients no longer diminish so quickly. The main difference between GRU and LSTM is that GRU has two gates, while LSTM has three gates. The gates are updated at each time step, and they determine how much information is passed from the previous time step to the current time step.GRUs are simpler than LSTMs, and they are often faster to train.

GRUs are a more complicated node structure than a base RNN in which the value of the previous state is scored for how important it is in the next state and only as much as is optimal is used to compute the next state. This helps alleviate the vanishing

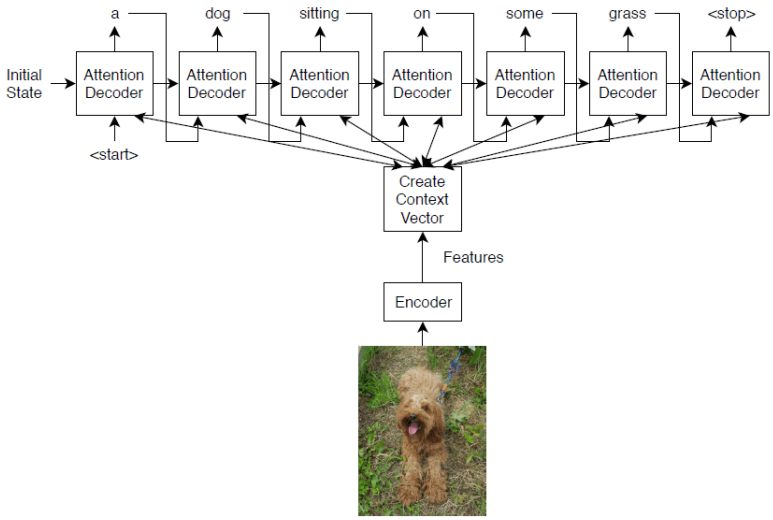

Fig. 3: Attention Mechanism

gradient problem as the values of the gradients are no longer exponentiated at each step in backpropagation and the gradients no longer diminish so quickly. The new equation for the next hidden state is :

$$
\begin{aligned}
z &= \sigma(State_{i-1} * H_z + Word_i * I_z + Image * N_z) + b_z \\
r &= \sigma(State_{i-1} * H_r + Word_i * I_r + Image * N_r) + b_r \\
\hat{h} &= tanh(r * State_{i-1} * H + Word_i * I + Image * N) + b \\
State_i &= z * State_{i-1} * (1 - z) * \hat{h}
\end{aligned}
$$

Here z is the update gate which is used to calculate how much of the new state should come from the previous state and how much of it should come from the new value that it is calculating, and r is the reset gate which calculates how much of the previous state should go into the new value that is being calculated, $\hat{h}$. All other values and matrices have the same values and dimensions as they did in the previous baseline model.

4.5 GRU Attention Model

I used the Bahdanau soft attention system [7]. This deterministic attention mechanism makes the model as a whole smooth and differentiable.

Attention is a mechanism that allows a model to focus on specific parts of an input. In image captioning, attention can be used to help the model focus on the most relevant parts of an image when generating a caption.

There are two main types of attention: hard attention and soft attention. Hard attention is a deterministic mechanism that always focuses on the same parts of an

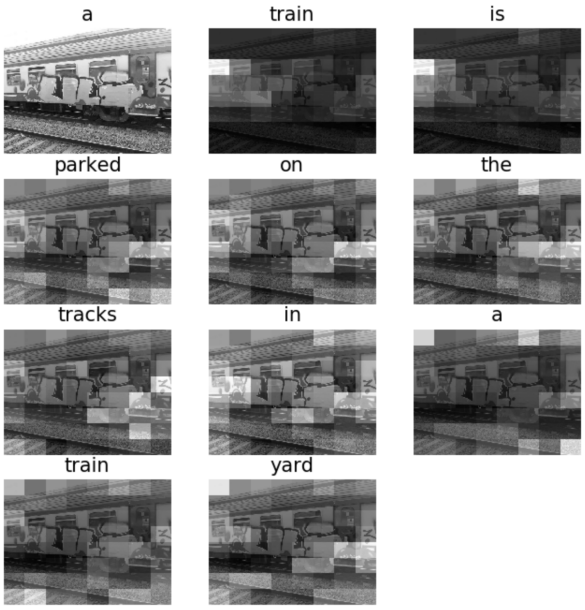

Fig. 4: Example of attending over an image to generate a caption.

image. Soft attention is a probabilistic mechanism that can focus on different parts of an image depending on the context.

My method uses soft attention. Soft attention is a more flexible and powerful mechanism than hard attention. It allows the model to learn how to focus on the most relevant parts of an image for each word in the caption.

Figure below shows how attention can be used for image captioning. The image is first encoded by a convolutional neural network (CNN). The CNN outputs a feature map that represents the important features of the image. The feature map is then passed to an attention layer. The attention layer calculates a weight for each part of the feature map. The weights are then used to create a weighted sum of the feature map. The weighted sum is then passed to a Gated Recurrent network (GRU). The GRU generates the caption one word at a time. The attention mechanism allows the GRU to focus on the most relevant parts of the image when generating each word.

The attention mechanism has been shown to be effective in a variety of image captioning tasks. It has been shown to improve the accuracy and fluency of the captions that are generated.

4.6 Evaluation

The model was scored by computing the perplexity, which is a measure of how well the sentence fits the language model and how well it works to describe the image that had been given to the caption.

The score of the language model was calculated based on how similar successive word vectors were to each other. Words that occur more closely to each other are more likely to have a similar value since the vectors are created based off their co-occurrences, so similar word vectors are more likely to have a low perplexity.

The score for how well the sentence fits the image was created by calculating how likely it is for the word to be selected based off of the image. To accomplish this, I went through the dataset and calculated the likelihood of each class of object is given that a given word is included in its caption. Then, to compute the perplexity based off how well each word fits the image, the likelihood of the class given a caption word is compared to the likelihood of the classes given the image. The equation for calculating the loss is as follows :

$$J(\theta) = -\sum_{i=1}^{||V||} y_i^{(t)} * log(y_i^{\hat{(t)}} - \sum_{j=1}^{||I||}\sum_{i=1}^{||I||} P(y_i^{(t)}|Image_j) * log(y_i^{\hat{(t)}})$$

I also calculated BLEU scores using the API provided by the Python API from the COCO site. BLEU is a metric that measures the similarity between a generated sentence and a reference sentence. The BLEU score is calculated by comparing the n-grams of the generated sentence to the n-grams of the reference sentence. The higher the BLEU score, the more similar the generated sentence is to the reference sentence.

5 Results

The perplexity and BLEU scores for each of the different models is shown in the following table:

Model	Perplexity	BlEU1	BlEU2	BlEU3	BlEU4
Baseline	80.40	66.7	51.1	39.9	29.92
GRU + GloVe	32.93	72.5	55.6	41.4	30.6
GRU Attention Model	26.42	73.7	57.1	43.1	32.4

The baseline model achieved a perplexity of 80.40, indicating a higher level of uncertainty and lower predictive power. In terms of BLEU scores, the baseline model showed moderate performance, with BLEU1 at 66.7, BLEU2 at 51.1, BLEU3 at 39.9, and BLEU4 at 29.92. These scores indicate that the generated captions partially overlap with the reference captions but lack precision and accuracy.

The GRU + GloVe model outperformed the baseline model in terms of perplexity, achieving a significantly lower value of 32.93. This indicates a better understanding and prediction of the captions. The BLEU scores also improved for this model, with BLEU1 at 72.5, BLEU2 at 55.6, BLEU3 at 41.4, and BLEU4 at 30.6. These scores indicate an enhanced level of overlap and accuracy compared to the baseline model.

The GRU Attention Model demonstrated the best performance among the three models. It achieved the lowest perplexity of 26.42, indicating a high level of understanding and prediction accuracy. The BLEU scores for this model were also the highest,

(a) Caption Generated by GRU Model

(b) Caption Generated by GRU Attention Model

Fig. 5: Comparison of Models

with BLEU1 at 73.7, BLEU2 at 57.1, BLEU3 at 43.1, and BLEU4 at 32.4. These scores suggest that the generated captions closely match the reference captions, showing improved precision and accuracy. Current state-of-the-art implementations scored and trained on the MS-COCO databases have scored a perplexity of 14.23. My value is comparable to that value and while mine is a little higher than is to be expected as I have had a much smaller amount of time and computing power and there were a decent amount of concessions I had to make in the interest of what was possible in the given time and resources

Based on these results, it can be concluded that the GRU Attention Model is the most effective approach for image captioning, yielding captions with higher quality and better alignment with the reference captions.

6 Future Work

One of the main ways to improve this model would be to train the convolutional neural network (CNN) along with the recurrent neural network (RNN). Since I did not have the time or computing power to back propagate the errors to the CNN I used the pre-trained Inception network. This can lead to CNN not being as accurate as it could be. If the CNN were trained along with the RNN, it would be able to learn from the errors that the RNN makes. This would lead to a more accurate and robust model.

Finally, the RNN could be made more complicated by using a multi-layer RNN. A multi-layer RNN is a type of RNN that has multiple layers of neurons. Each layer of neurons learns a different representation of the input data. This allows the RNN to learn more complex relationships between the input data and the output data. A multi-layer RNN would likely lead to a more accurate and robust model.

References

[1] Wang, C., Yang, H., Bartz, C., Meinel, C.: Image captioning with deep bidirectional lstms. In: Proceedings of the 24th ACM International Conference on Multimedia,

pp. 988–997 (2016)

[2] Kulkarni, G., Premraj, V., Ordonez, V., Dhar, S., Li, S., Choi, Y., Berg, A.C., Berg, T.L.: Babytalk: Understanding and generating simple image descriptions. IEEE transactions on pattern analysis and machine intelligence **35**(12), 2891–2903 (2013)

[3] Yatskar, M., Galley, M., Vanderwende, L., Zettlemoyer, L.: See no evil, say no evil: Description generation from densely labeled images. In: Proceedings of the Third Joint Conference on Lexical and Computational Semantics (* SEM 2014), pp. 110–120 (2014)

[4] Li, C., Harrison, B.: Stylem: Stylized metrics for image captioning built with contrastive n-grams. arXiv preprint arXiv:2201.00975 (2022)

[5] Zhang, W., Shi, H., Tang, S., Xiao, J., Yu, Q., Zhuang, Y.: Consensus graph representation learning for better grounded image captioning. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, pp. 3394–3402 (2021)

[6] Microsoft COCO Dataset. https://cocodataset.org/#download,

[7] Bahdanau, D., Cho, K., Bengio, Y.: Neural Machine Translation by Jointly Learning to Align and Translate (2016)