image captioning

ahmadreza baqerzadeh

October 2023

introduction

Image captioning refers to the task of generating a descriptive and meaningful caption for an image. The goal is to develop a model or algorithm that can analyze the visual content of an image and generate a concise and semantically relevant description that captures the key elements and context of the image. The challenge lies in understanding the complex visual information, extracting relevant features, and generating linguistically coherent and contextually appropriate captions that align with human perception and understanding. The image captioning problem requires a combination of computer vision techniques, natural language processing, and deep learning to bridge the gap between visual and textual understanding. There are several approaches or ways to tackle image captioning:

1. Convolutional Neural Networks (CNN) + Recurrent Neural Networks (RNN): This approach combines the power of CNNs for image feature extraction and RNNs, such as Long Short-Term Memory (LSTM), for generating captions. The CNN is used to extract visual features from the image, which are then fed into the RNN to generate a sequence of words constituting the caption.

2. Attention Mechanism: In this approach, an attention mechanism is introduced to enhance the caption generation process. The attention mechanism allows the model to focus on different parts of the image while generating the caption, giving more relevant and context-aware descriptions.

3. Transformer Networks: Inspired by the success of Transformer models in natural language processing tasks, Transformer-based architectures have been applied to image captioning. These models utilize self-attention mechanisms to capture dependencies between image regions and words, enabling more effective caption generation.

4. Reinforcement Learning: Some approaches incorporate reinforcement learning techniques to optimize the captioning process. Reinforcement learning methods use reward-based feedback to iteratively improve the generated captions, making them more accurate and coherent.

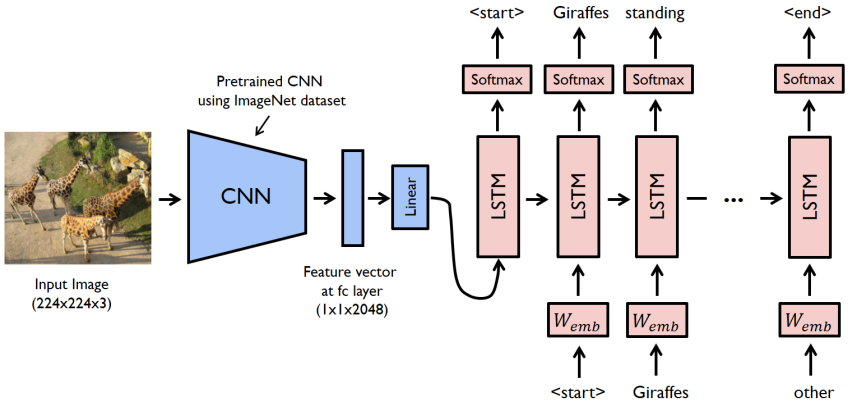
5. Multimodal Approaches: In this approach, the visual information from the image is combined with textual information, such as pre-trained word embeddings or linguistic features. This multimodal fusion enables a better understanding of the image and improves the quality of the generated captions.

6. Transfer Learning: Utilizing pre-trained models trained on large-scale datasets, such as ImageNet, can provide a head start for image captioning tasks. By transferring the learned visual features to the captioning model, it can benefit from the generalization and feature representation capabilities of the pre-trained model.

It's important to note that the choice of approach depends on the specific requirements of the image captioning task and the available resources. Different methods have their strengths and weaknesses, and ongoing research aims to further advance the field of image captioning.

Model

The **CNN+LSTM** model is a popular approach for image captioning tasks. The Convolutional Neural Network (CNN) is used for extracting visual features from the input image, while the Long Short-Term Memory (LSTM) network is employed to generate a coherent and descriptive caption based on those features.



In this architecture, the CNN component is responsible for extracting visual features from the input image. By applying a series of convolutional and pooling layers, the CNN can capture spatial information and high-level features that represent the content of the image. These visual features serve as meaningful representations of the image and are fed into the LSTM network.

The LSTM network, being a recurrent neural network, processes the visual features sequentially while generating the textual captions. At each time step, the LSTM takes the previous hidden state and the current visual feature as inputs and produces a new hidden state and a prediction for the next word in the caption. This process is repeated until a complete caption is generated. The LSTM's ability to retain memory of previous states allows it to capture the temporal dependencies necessary for coherent caption generation.

The CNN component of the model helps to capture the spatial information and high-level features of the image, allowing it to understand the content and context. The extracted visual features serve as the input for the LSTM network.

The LSTM network is responsible for generating the textual captions. It sequentially processes the visual features and generates words or phrases one by one, considering the context and incorporating the information from previously generated tokens. The recurrent nature of the LSTM enables it to capture the temporal dependencies in the caption generation process.

The combination of CNN and LSTM in an end-to-end architecture has shown promising results in image captioning tasks. It allows the model to effectively leverage both the visual and textual information, producing more accurate and meaningful captions for images.

Dataset

The **Flickr8K** dataset is a popular and widely used benchmark dataset for image captioning tasks. It was created by collecting images from the photo-sharing website Flickr and pairing them with descriptive captions written by human annotators.

Here are some key details about the Flickr8K dataset:

1. Size and Contents: The dataset consists of 8,000 images, hence the name "Flickr8K." Each image is accompanied by five different captions, resulting in a total of 40,000 captions. The images cover a wide range of topics, objects, scenes, and activities, making the dataset diverse and suitable for training and evaluating image captioning models.

2. Caption Quality: The captions in the dataset were manually written by human annotators, which ensures a good quality and relevance to the corresponding images. These captions effectively describe the visual content and provide useful context for the image captioning models to learn from.

3. Language Style: The captions in the Flickr8K dataset are written in a more formal and descriptive style, reflecting the type of captions typically found in photo collections or image databases. This style of captions makes the dataset suitable for certain applications or scenarios where formal descriptive language is preferred.

4. Split: The dataset is divided into predefined train, validation, and test splits. The train split contains 6,000 images with their corresponding captions, while the validation and test splits each consist of 1,000 images with their respective captions. This split allows researchers to train and evaluate image captioning models in a standardized manner.

Due to its size, diversity, and well-annotated captions, the Flickr8K dataset has been extensively used by researchers and practitioners as a benchmark for developing and evaluating image captioning models. It serves as a valuable resource for training and testing the performance of state-of-the-art image captioning algorithms, allowing for fair comparisons and advancements in the field. Here are two examples:

1) A brown dog chases something a man behind him threw on the beach.
2) A man and a dog on the beach.
3) A man is interacting with a dog that is running in the opposite direction.
4) A man playing fetch with his dog on a beach.
5) A man walking behind a running dog on the beach.

1) A man does skateboard tricks off a ramp while others watch.
2) A skateboarder does a trick for an audience.
3) Boy dressed in black is doing a skateboarding jump with a crowd watching.
4) Two dogs on pavement moving toward each other.
5) People watching a guy in a black and green baseball cap skateboarding.

Use CNN+LSTM

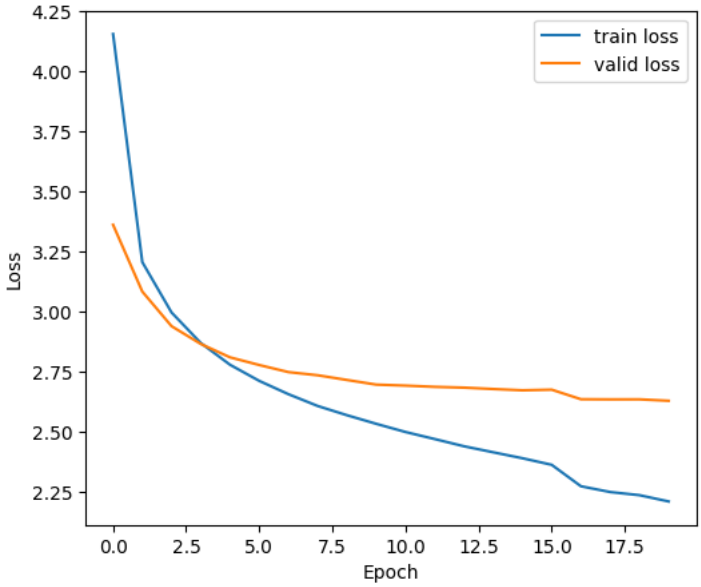
In this stage, I first separated the dataset based on the captions, resulting in a total of **30,000** samples. Specifically, we have **6,000** images for the training set, with each image having **5 captions**. Therefore, the total number of training samples is **30,000**. In the validation set, we have **1,000** images, and considering five captions per image, the total number of samples in this set is **5,000**. The test dataset consists of **approximately 5,000** samples (some images in the test set have more than 5 captions). Overall, the entire dataset consists of **40,460 captions**, which is equal to the total number of samples.

I attempted to train the model in two different modes, considering the total number of tokens in the dataset and the number of tokens in the training data. The training set has a token count of 7,698 (excluding special tokens), and the total token count for the training data is 8,911. In the first section, which includes a simple convolutional neural network and a recurrent neural network, I only use the tokens from the training set. For the convolutional network, I utilize **ResNet50**, where the network is frozen during the training process, and its parameters are not updated. It should be noted that the last layer of ResNet50, which is a linear layer, needs to be modified and incorporated during the training process. The output of this network should be combined with the output of the embedding layer, and the size or the last dimension of the output should match the embedding-dim. After combining the outputs of the convolutional network and the embedding-layer, I concatenate them in a way that the output of the convolutional layer, which represents the image features, behaves like a token and is added to the beginning of the caption tokens. When combining these two outputs, it's important to note that the last token of the caption is not considered, and similar to language modeling, the network needs to receive the image and generate a caption with the same size and dimensions as the actual caption. Therefore, the last token is not added to them. Additionally, to create the dataset class, it's necessary to consider that in order for the captions to concatenate row-wise in a batch, the size of the longest sequence should be considered, and the other captions should be padded to be able to concatenate them. After combining the caption tokens and the feature vector obtained from the image, they pass through LSTM layers to generate the output, and finally, for classification, they pass through a linear layer to adjust the output size. The hyperparameters of the best model are as follows:

embedding-dim: 256
hidden-dim: 256
num-layers: 2
embedding-dropout: 0.3
lstm-dropout: 0.3
batch-size: 128
learning rate: 0.9
weight decay: 1e-6
optimizer : SGD

. Note that the training loop is configured in such a way that if the validation loss does not improve in each epoch, the learning rate is **reduced to 0.2**, and the optimizer continues its work with the new learning rate.

Result



parameters	loss on valid	loss on test
5.52M	2.62	2.62

	BLEU valid	BLEU test
n_gram = 1	0.499705	0.495730
n_gram = 2	0.263855	0.262642
n_gram = 3	0.133867	0.135254
n_gram = 4	0.073006	0.076975

Captioning

```
'a boy in a red shirt is jumping on a trampoline .'
```

```
image_pil
```

```
'a man is surfing in the ocean .'
```

```
image_pil
```

Add Attension

Adding attention to a CNN+LSTM model for image captioning is a common and effective approach that improves the model's ability to generate more accurate and contextually relevant captions. In the standard CNN+LSTM architecture, the CNN extracts visual features from the image, which are then passed to the LSTM to generate the caption. However, this approach treats all image regions equally and may not focus on the most informative parts of the image when generating each word of the caption.

Attention mechanisms address this limitation by allowing the model to dynamically focus on different regions of the image while generating each word of the caption. It learns to assign different weights or importance to different regions of the image based on their relevance to the current word being generated. This enables the model to attend to relevant image regions and generate more contextually accurate captions.

The attention mechanism typically involves adding an attention layer between the CNN and LSTM components of the model. This layer takes the visual features from the CNN and the hidden state of the LSTM at the current time step as input. It calculates attention weights for different image regions based on the similarity between the visual features and the LSTM hidden state. These attention weights are then used to compute a weighted sum of the visual features, which is combined with the LSTM hidden state to generate the next word of the caption.

By incorporating attention, the model gains the ability to focus on relevant image regions, providing more contextually meaningful information to guide the caption generation process. This can lead to improved performance in generating captions that accurately describe the content of the image. The attention mechanism has been widely adopted in state-of-the-art image captioning models and has shown significant improvements in caption quality and relevance.

"Show, Attend and Tell: Neural Image Caption Generation with Visual Attention" is a research paper published in 2015 by Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard Zemel, and Yoshua Bengio. The paper introduces an attention-based model for image captioning that significantly improves the generation of descriptive and accurate captions.

The authors' approach, known as the Show, Attend and Tell (SAT) model, combines a convolutional neural network (CNN) for image feature extraction and a long short-term memory (LSTM) network for caption generation, incorporating attention mechanisms to focus on relevant image regions.

In the SAT model, the CNN processes the input image and extracts a set of visual features. These features are then used to initialize the LSTM network, which generates captions word by word. Crucially, the model incorporates attention mechanisms to dynamically attend to different image regions while generating each word of the caption.

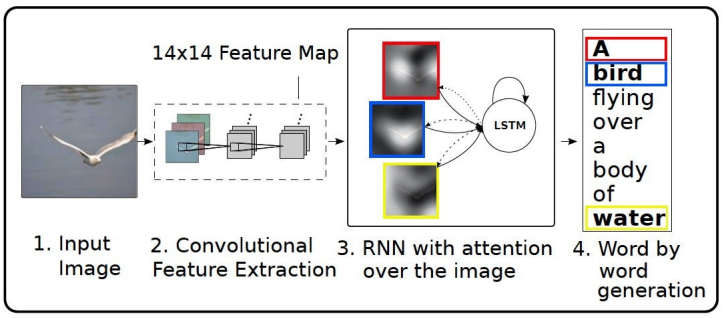
The attention mechanism in SAT is based on soft attention, where the model learns to assign weights or probabilities to different image regions at each time

step of caption generation. These weights indicate the importance or relevance of each region for generating the current word. The attention weights are computed by comparing the visual features with the LSTM hidden state, allowing the model to focus on relevant regions.

During the caption generation process, the SAT model attends to different image regions, providing contextually relevant information for generating each word. The attention weights are used to compute a weighted sum of the visual features, which is combined with the LSTM hidden state to predict the next word.

The paper demonstrates that the SAT model outperforms previous state-of-the-art approaches in image captioning, achieving significant improvements in caption quality and relevance. The attention mechanism allows the model to selectively focus on relevant image regions, leading to more accurate and contextually meaningful captions.

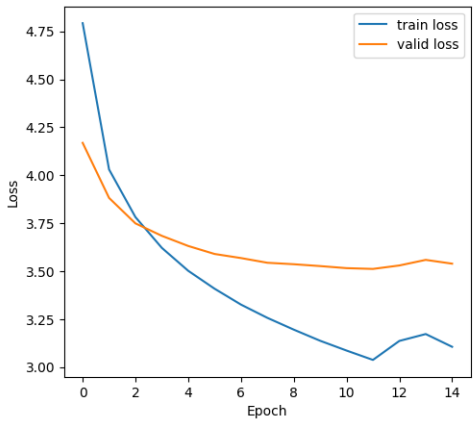
The SAT model has had a profound impact on the field of image captioning and paved the way for subsequent research on attention-based models in various natural language processing and computer vision tasks. It highlights the importance of incorporating attention mechanisms to enhance the performance of neural models in tasks involving sequential data and complex visual information.

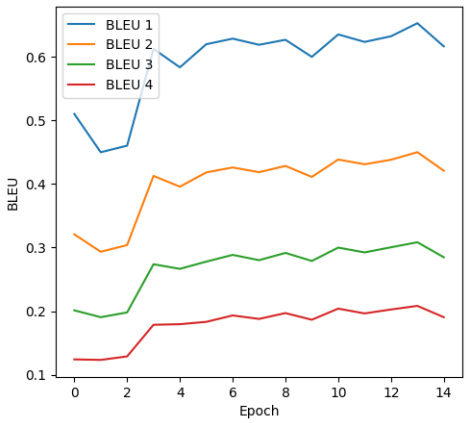


The hyperparameters of the best model are as follows:
attention-dim : 256
embed-dim : 512
decoder-dim : 512
dropout : 0.5
batch-size: 32
learning rate for decoder: 4e-4
weight decay for decoder: 1e-5
optimizer : Adam

Note that the training loop is configured in such a way that if the validation loss does not improve in each epoch(if epoch number greater than 10), the learning rate is **reduced to 0.8**, and the optimizer continues its work with the new learning rate.

Result





parameters	loss on valid	loss on test
19.23M	2.77	2.77

	BLEU Valid	BLEU Test
n_gram = 1	0.654036	0.651990
n_gram = 2	0.451890	0.446859
n_gram = 3	0.308973	0.305348
n_gram = 4	0.208058	0.207636

Captioning

'a surfer is riding a wave .'

image_pil

'a man stands on a rock overlooking the mountains .'

image_pil

```
'a black and white dog is running on the beach .'
```

```
image_pil
```

```
'a man is riding a bike in the air .'
```

```
image_pil
```

some details

Version of libraries:
numpy : 1.23.5
torch : 2.1.0+cu118
torchtext : 0.16.0+cpu
torchvision : 0.16.0+cu118
tqdm : 4.66.1