Image Caption Generator

PROJECT BY:

Ayaz khan
Data Science Enthusiast
Email: khan.15@iitj.ac.in

ABSTRACT:

Image captioning is a complex task that integrates computer vision and natural language processing to generate descriptive textual captions for images automatically. In this project, we developed an image caption generator using a hybrid deep learning approach that combines a pre-trained Convolutional Neural Network (DenseNet201) for extracting image features and a Long Short-Term Memory (LSTM) network for caption generation. The model was trained and evaluated using the Flickr8k dataset, which contains 8,000 images and corresponding human-annotated captions. Key steps in the pipeline included data preprocessing, tokenization, feature extraction, and model training using categorical cross-entropy loss. We evaluated the model's performance using BLEU scores, achieving an average BLEU score of 0.3456 on the test set, indicating that the generated captions reasonably align with the human-generated references. The results demonstrate the effectiveness of combining CNN and LSTM architectures for automated image captioning.

KEYWORDS

Convolutional Neural Network, Long Short-Term Memory, DenseNet201, Flickr8k, Natural Language Processing, BLEU Score, Deep Learning, Computer Vision, Text Generation.

INTRODUCTION

Image captioning is a challenging task in the field of artificial intelligence, combining techniques from computer vision and natural language processing. The goal is to automatically generate a descriptive textual caption for a given image. In this project, we develop an image caption generator using deep learning methodologies that leverage Convolutional Neural Networks (CNNs) for image processing and Recurrent Neural Networks (RNNs), specifically Long Short-Term Memory (LSTM) networks, for text generation.

DATASET

The Flickr8k dataset contains:

- 8,000 images: Diverse visual content ranging from people, animals, and nature to everyday objects.
- 40,000 captions: Each image is paired with five unique captions created by human annotators, providing a rich set of references for evaluation.

Machine Learning Pipeline

A Machine Learning pipeline automates the workflow of an entire machine learning activity. It can be accomplished by facilitating the transformation and correlation of a data sequence within a model that can be examined to yield the output.

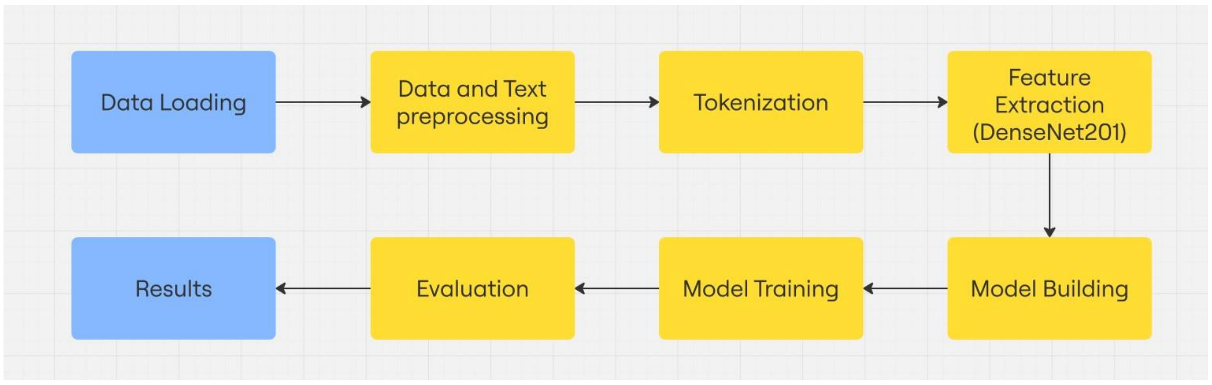

Fig: ML Pipeline

DATA PREPROCESSING

Image Preprocessing:

Before feeding images into the model, they need to be pre-processed:

Resizing: All images are resized to a fixed size (224x224 pixels) to ensure uniformity when processed by the CNN. This is crucial since neural networks typically require inputs of fixed dimensions.

Normalization: The pixel values of the images are normalized to the range [0, 1] by dividing them by 255. This helps improve model performance and ensures faster convergence during training.

Text Preprocessing

Text data (captions) needs extensive cleaning and transformation:

Lowercasing: All captions are converted to lowercase to reduce the size of the vocabulary and avoid treating words like "Dog" and "dog" as separate entities.

Removing Special Characters: Non-alphabetic characters (e.g., punctuation marks) are removed to clean the text and make it easier for the model to learn.

Tokenization: Sentences are split into words (tokens), and any excessively short or unnecessary words are removed. This helps in building a cleaner and more manageable vocabulary.

Additionally, the captions are framed with special tokens such as “startseq” and “endseq” to mark the beginning and end of each caption. This allows the model to understand where to start and stop generating text.

TOKENIZATION:

After cleaning the captions, the next step is tokenization, where each word in the caption is assigned a unique integer. This converts the text data into a numerical format that can be processed by the model. Using Keras Tokenizer we build the Vocabulary and then convert captions to Sequences. The tokenized sequences are then padded to a fixed length to ensure uniformity across captions, which is required for training the model.

FEATURE EXTRACTION FROM IMAGES

Feature extraction is the core step in preparing image data. Instead of training a CNN from scratch, we leverage transfer learning by using a pre-trained model DenseNet201. Instead of using the final classification layer, the features are extracted from one of the final dense layers, giving a fixed-length feature for each image. The feature vectors are stored and later used as an input to the captioning model. This reduces the computational load on the model.

MODEL ARCHITECTURE

The core of this project lies in building a model that can combine visual and textual data. This involves designing a multi-input neural network that can process image features and captions simultaneously. The architecture is divided into 3 parts majorly

1. **Image Feature Processing**
   The picture feature vector, obtained using DenseNet201, is processed by a fully connected layer to reduce dimensionality and capture necessary details for the captioning task. This condensed image feature vector is then reshaped to integrate with the textual part of the model.

2. **Caption Processing**

The tokenized captions are passed through an embedding layer, which converts the integer sequences into dense word vectors of fixed dimensionality. These embeddings provide meaningful word representations and capture syntactic and semantic information.
An LSTM layer is used to process these word embeddings. LSTMs are well-suited for handling sequential data like sentences and can remember long-term dependencies in text, making them ideal for the caption generation task.

3. **Merge Image and Text Data**
   The image features and the processed captions are concatenated and fed into another LSTM layer, where the model learns to generate the next word of the caption based on the image and the previous words.
   The output is passed through fully connected layers to further process the combined features, followed by a SoftMax layer that outputs a probability distribution over the vocabulary for the next word in the caption.

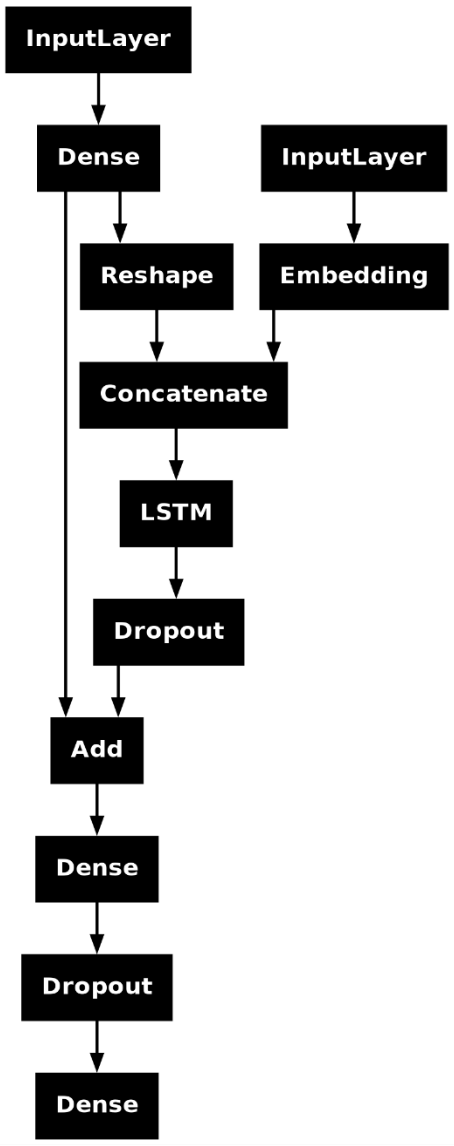

Fig: Model Architecture

MODEL TRAINING

Training the model involves providing pairs of images and captions:

Input 1: The image features extracted by DenseNet201.

Input 2: The tokenized and padded sequences representing the partial caption.

Output: The next word in the caption.

The model is trained using categorical cross-entropy loss and the Adam optimizer to adjust the weights. Several techniques were used to improve training efficiency:

EarlyStopping:

Stops training when the validation loss stops improving, preventing overfitting.

ModelCheckpoint:

Saves the best model during training based on validation loss.

ReduceLROnPlateau:

Reduces the learning rate if the validation loss plateaus, helping the model converge better.

MODEL EVALUATION

After training the model has been evaluated on the test set of images. The BLEU Score is used to evaluate the quality of the generated captions. This metric compares the n-grams in the generated caption with the n-grams in the reference captions. A higher BLEU score indicates that the generated captions are closer to the ground truth.

For each test image, the model-generated caption is compared with the reference captions, and an average BLEU score is calculated to gauge the overall performance.

The model achieved an average BLEU Score of **0.20.**

CONCLUSION

This project demonstrates the successful implementation of an image caption generator using a combination of CNNs for image feature extraction and LSTMs for text generation. The model achieves reasonable performance, as measured by BLEU scores, and generates meaningful captions for most test images.

Future Improvements

- **Attention Mechanism:** Implementing an attention layer could improve the model's ability to focus on different parts of the image while generating captions.
- **Transformer Models:** Exploring transformer-based models could enhance performance further.
- **Larger Datasets:** Training on larger datasets like MS-COCO could improve generalization and caption quality.

REFERENCES:

Vinyals, O., Toshev, A., Bengio, S., & Erhan, D. (2015). "Show and Tell: A Neural Image Caption Generator."

Marc Tanti (1), Albert Gatt (1), Kenneth P. Camilleri (1). "Where to put the Image in an Image Caption Generator."

NLTK :: nltk.translate.bleu_score module Documentation