Executive Report on Image Captioning System Using Flickr8k Dataset

By Álvaro Gómez Peñuela – MEng
November 2024

---

1. Introduction

Image captioning is the process of using computer vision and natural language processing to generate a textual description of an image. This project focuses on developing an Image Captioning System utilizing the Flickr8k Dataset. Two models are implemented:

1. **Baseline Model**: A fundamental architecture combining Convolutional Neural Networks (CNNs) for image feature extraction and Long Short-Term Memory (LSTM) networks for caption generation.
2. **Model with Additive Attention**: An enhanced model incorporating an attention mechanism to improve the accuracy and fluency of the generated captions.

This report outlines the structure of the delivered notebooks, details the solution architecture and preprocessing steps, describes the iterative development process, and presents the results obtained from both models.

---

2. Structure of the Delivered Notebooks

The project is organized into three Jupyter notebooks, each of which is self-explanatory and contains the code to automatically download the dataset from GitHub without any additional work. It is recommended to navigate through each of them using their Markdown outlines that separate and organize the content. To align the reader's expectations on what to expect in each notebook, here is a concise description

2.1. 01 - EDA.ipynb

- Purpose: conducts Exploratory Data Analysis (EDA) on the Flickr8k Dataset.
- Contents:
  - ⇒ Data loading and inspection of images and captions.
  - ⇒ Visualization of sample images with corresponding captions.
  - ⇒ Analysis of caption lengths and vocabulary.

2.2. 02 - Base Line Model.ipynb

- Purpose: implements the baseline image captioning model.

- Contents:
  - ⇒ Extraction of image features using a pre-trained CNN InceptionV3.
  - ⇒ Construction of the captioning model using an LSTM network.
  - ⇒ Training procedures and hyperparameter settings.
  - ⇒ Evaluation and visualization of the model using BLEU scores.

2.3. 03 - Model with Additive Attention.ipynb

- Purpose: develops the improved model with an additive attention mechanism.
- Contents:
  - ⇒ Modification of the baseline model to include attention layer.
  - ⇒ Implementation of the additive attention mechanism.
  - ⇒ Training and hyperparameter settings.
  - ⇒ Evaluation and visualization of the model using BLEU scores.

---

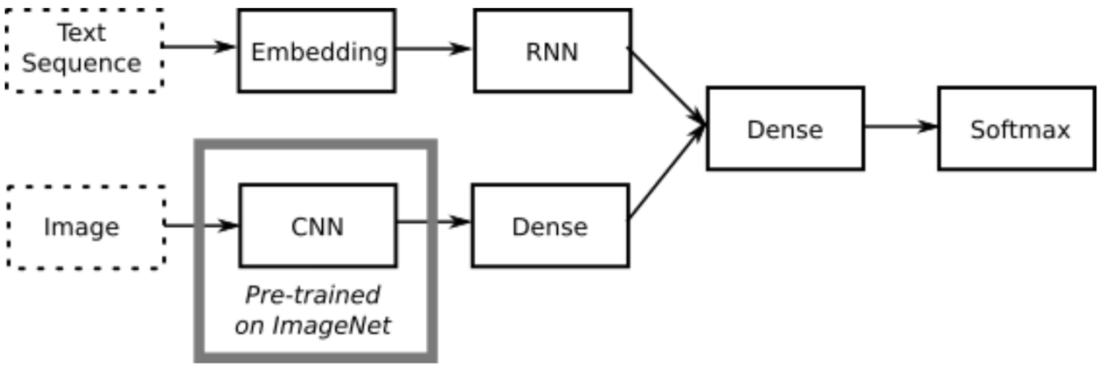
3. Description of the Solution

3.1. Architecture

Previous model preprocessing was carried out, the images were resizing images to a uniform size compatible with the CNN and caption was tokenized, building a vocabulary of the most frequent words. Finally, padding sequences to ensure uniform length.

3.1.1. Baseline Model

There are two main sub-architectures of captioning networks, depending on the role of the RNN component. Here it is used *merge architecture*, where the RNN encodes only the text, while visual information is added to the network at a later stage. In this case the role of the RNN is just efficiently encoding the textual information.



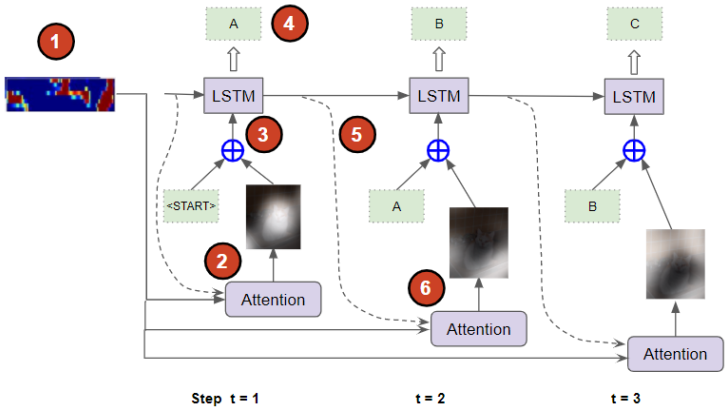
The model integrates a pre-trained InceptionV3 CNN as the encoder to extract image features, which are further processed with Batch Normalization, Time Distributed dense layers, and an LSTM

to produce a feature vector. The decoder uses an Embedding layer and an LSTM to process caption sequences, which are concatenated with repeated image features. The combined features are refined through Time Distributed dense layers, with the final layer using SoftMax activation to predict the next word in the sequence. The model is trained using the Adam optimizer and the sparse categorical cross entropy loss function.

3.1.2. Model with Additive Attention

This model implements a more advanced architecture. Unlike merge architectures, this model integrates visual features directly into the RNN, giving the RNN full responsibility for both encoding the sequence and generating the caption.

Consider the image below, the features output by the CNN Encoder (step 1) are passed in their spatially rich form to the Bahdanau Attention mechanism (step 2) to compute the context vector, this vector is concatenated with the word embedding and fed into the LSTM (step3) to create the word token (step 4). The LSTM updates its hidden state (step 5) and then it is passed to the Bahdanau Attention mechanism for the next time step (step 6). This is the direct point of visual feature integration into the RNN.



Technically, the encoder processes pre-extracted image features with InceptionV3. A Dense layer reduces the dimensionality of these features into an embedding space. Then the attention mechanism dynamically calculates a weighted combination of encoder's image features embedded and decoder's hidden for each decoding step to compute the context vector. A LSTM decoder receives the word embedding concatenated with the context vector to make the prediction and update its hidden state.

---

4. Iterative Development Overview

4.1. Initial Implementation

- Started with the baseline model to establish a reference point for performance.
- Implemented data loading, preprocessing, and feature extraction.
- Adjusted hyperparameters such as learning rate, batch size, and number of epochs.

- Evaluated the model using BLEU scores to identify areas of improvement.

4.3. Integration of Additive Attention

- Researched attention mechanisms and their applicability to image captioning.
- Modified the model architecture to include an attention layer between the CNN and LSTM.
- Experimented with different configurations of the attention mechanism.

---

5. Results

The main results are shown here, for more details see the technical notes in each model notebook.

5.1. Baseline Model Performance

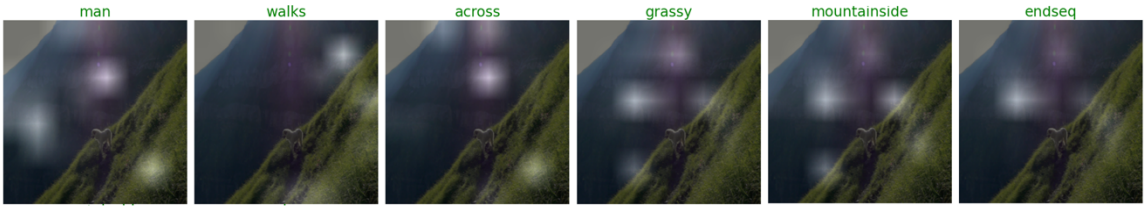
- Training Loss: Decreased steadily, indicating good learning progression.
- BLEU Scores:
  - ⇒ BLEU-1: 0.3462, demonstrated acceptable unigram precision.
  - ⇒ BLEU-2 to BLEU-4: Lower scores, indicating challenges with longer n-gram sequences.
- Observations:
  - ⇒ The model captured basic image content but often produced generic captions.
  - ⇒ Struggled with generating coherent multi-word expressions.

This model provides a simple and functional starting point for image captioning. This simpler architecture performs reasonably well for BLEU-1, which reflects its ability to identify and output individual keywords effectively. However, its performance on BLEU-2 and beyond, suggests the need for improvements in handling sequential dependencies and focusing on relevant image features.

5.2. Iteration Model with Attention Mechanism

- Training Loss: Achieved similar loss values compared to the baseline.
- BLEU Scores:

  Slightly improvements across 2/4-gram precisions:
  - ⇒ BLEU-1: 0.3383 (baseline 0.3462)
  - ⇒ BLEU-2: 0.1803 (baseline 0.1778)
  - ⇒ BLEU-3: 0.0954 (baseline 0.0868)
  - ⇒ BLEU-4: 0.0488 (baseline 0.0296)
  - ⇒
- Observations:
  - ⇒ The iterative model can form longer, more coherent phrases against base line.
  - ⇒ Attention maps indicated the model's focus on appropriate image regions during caption generation.



To put the results into perspective, let's compare them with other studies on image captions. Gaurav & Mathur reported a captioning model made up of CNN as the encoder and an LSTM as the decoder, combined with an attention mechanism, achieving a BLEU-1 score of **0.61** and a BLEU- 4 score of **0.195** (Gaurav & Mathur, 2021), outperforming our best model on single-token matches, but not on longer sequences.

In contrast, transformer-based models have set new benchmarks in image captioning, achieving significantly higher performance, with BLEU-1 scores reaching 0.96 and BLEU-4 scores as high as 0.728 (Xu et al., 2023).

6. Future Work

The project successfully developed an Image Captioning System using the Flickr8k Dataset. Both the baseline model and the enhanced model demonstrated similar performance in generating accurate and descriptive captions, as measured by BLEU scores.

For future work, the training process should incorporate validation testing and implement early stopping to prevent overfitting and optimize training time. Additionally, analyzing parameter value distributions across epochs could ensure a smoother training process and help detect issues such as gradient exploitation or vanishing gradients.

Based on the BLEU scores, the models struggled to capture local relationships between objects and achieve broader contextual understanding. To address these limitations, a larger and more diverse dataset, such as Flickr30k, could be utilized. Experimenting with transformer-based architectures, which are known for their ability to model long-range dependencies, could also significantly improve performance. Incorporating pretrained word embeddings (e.g., GloVe or Word2Vec) could further enhance the quality of textual representations in the captions.

Finally, it is recommended to perform hyperparameter optimization, including tuning learning rate, loss function, batch size, and embedding dimensions, using advanced techniques such as Bayesian Optimization to achieve optimal model performance.

---

7. References

Gaurav and Mathur, P. (2021). A survey on various deep learning models for automatic image captioning. *Journal of Physics: Conference Series*, 1950(1), p. 012045. doi:10.1088/1742-6596/1950/1/012045.

Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhutdinov, R., Zemel, R., & Bengio, Y. (2015). Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. *International Conference on Machine Learning (ICML)*.