Image Captioning Using ResNet-50 and LSTMs

Alexander Holland
University of Victoria
Victoria, Canada
alexanderwholland@uvic.ca

Bhavanvir Rai
University of Victoria
Victoria, Canada
brai@uvic.ca

Kevin, Mathew
University of Victoria
Victoria, Canada
kevingmathew@uvic.ca

***Abstract*—This report outlines how neural networks could be used to predict the captions of images based on the feature of an image. The dataset for this project was collected from Flickr8k collection which consisted of images of over 8,000 images and over 40,000 image captions for the images [1] . After the image pre-processing was completed, the data was trained on a long short-term memory networks (LSTM) and a more sophisticated residual neural network (ResNet). The results revealed that neural networks are capable of predicting such features. However, to produce highly accurate predictions, the models must be fine-tuned, and the dataset needs to be large, diverse and optimized for the job.**

I. Introduction

Image captioning models aim to bridge the gap between visual content and natural language by generating descriptive captions for images. These models combine advancements in computer vision and natural language processing to automatically analyze the visual features of an image and generate coherent and contextually relevant textual descriptions. Typically, image captioning models employ deep learning architectures, such as convolutional neural networks (CNNs) for visual feature extraction and recurrent neural networks (RNNs), such as long short-term memory (LSTM) networks, for generating sequential captions. More recently, transformer-based models, such as the vision-language models, have gained prominence due to their ability to capture global image context and semantic relationships. The development of these image captioning models has revolutionized various domains, including accessibility, content indexing, and contextual understanding, opening up new possibilities for leveraging visual content in a textual form.

Task. Image captioning revolves around the challenge of automatically generating descriptive and contextually relevant captions for images. This task combines the fields of computer vision and natural language processing, aiming to bridge the gap between visual content and textual understanding. With the exponential growth of digital imagery across various domains, the ability to automatically generate captions for images has become increasingly valuable. Image captioning models leverage deep learning techniques, such as convolutional neural networks (CNNs) for visual feature extraction and recurrent neural networks (RNNs) or transformer-based models for generating sequential captions. These models learn to associate visual information with textual descriptions, allowing them to generate captions that capture the salient objects, actions, and relationships present within an image. The ultimate goal of image captioning is to enable machines to comprehend and describe visual content, thereby enhancing accessibility, content indexing, and contextual understanding across a wide range of applications. This report explores the methodologies, challenges, and advancements in image captioning, shedding light on the capabilities and potential impact of this task in the realm of computer vision and natural language processing.

II. Questions

1) What kind of model can we use to effectively tackle this problem?
2) What kind of data set could we use to train the model?
3) How do we measure the effectiveness of the model?
4) What are the practical applications of this project?

III. Analysis

Data analysis before creating an image captioning model is crucial. It provides insights into dataset characteristics, such as image distribution, caption lengths, and vocabulary. This information informs model design, addressing specific challenges and biases. By understanding the data, we can preprocess it appropriately and develop more accurate and relevant captioning models.

A. *Exploratory Data Analysis*

Dataset. The Flickr 8K dataset consists of 8,000 images paired with five captions each, resulting in a total of 40,000 captions. The images cover a wide range of subjects and scenarios, but no specific categories or tags are available. Analyzing the captions, we find that they vary in length, with an average of approximately 10 to 15 words. The minimum caption length is three words, while the maximum reaches around 20 words.

The vocabulary in the dataset is rich and diverse, with an estimated size of 8,000 to 10,000 unique words. Common words such as ”a,” ”the,” ”on,” ”in,” and ”and” indicate the presence of generic captions describing basic elements of the images. To assess the diversity and quality of the dataset, a random sample of images and their associated captions was examined, providing insights into the various image-caption pairs.

Image Sampling. Sampling images with their respective captions in the Flickr 8K dataset is crucial for evaluating caption quality, assessing alignment between visuals and text, identifying challenges, and understanding distribution patterns. By randomly selecting images and examining their associated captions, the quality and diversity of the dataset can be assessed, providing insights into descriptive power and creativity. Aligning images with captions enables the evaluation of caption accuracy and relevance, ensuring meaningful descriptions. Sampling also helps identify issues such as inconsistencies or repetition, guiding improvement strategies. Furthermore, understanding distribution patterns aids in modeling decisions, ensuring the system handles various image types effectively. In figure 1 we can see the sample images and their captions.

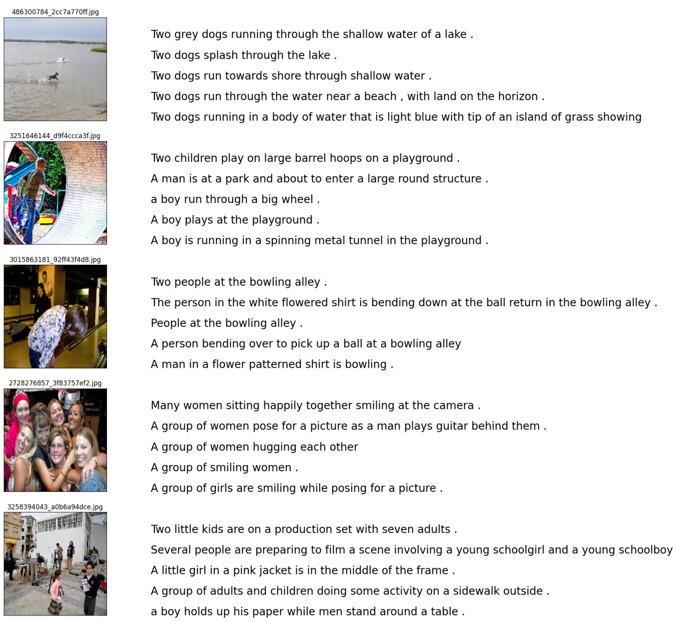

Fig. 1. Sample images with their 5 respective captions.

Word frequency. By examining the distribution of word occurrences in the caption dataset, we can gain valuable insights into the common vocabulary used. This analysis helps us identify frequently occurring words, such as articles and prepositions, which can guide preprocessing steps like stop-word removal. Additionally, it allows us to identify important and domain-specific terms that are relevant to the image captioning task. Understanding word frequency enables us to make informed decisions regarding vocabulary size, word embeddings, and language modeling techniques, ultimately improving the accuracy and coherence of generated captions.

We observed in Figure 2 that the most frequent words

Caption length. Analyzing the length of captions can offer insights into determining an optimal balance between

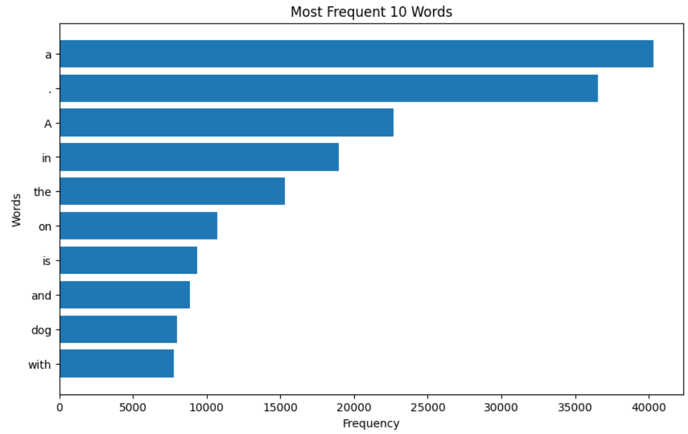

Fig. 2. Bar graph of the 10 most frequent words.

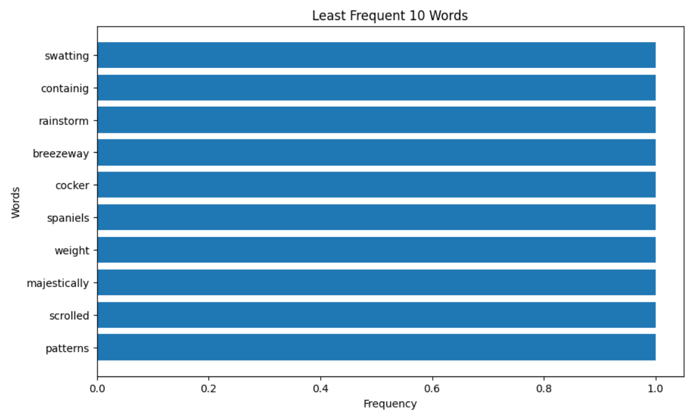

Fig. 3. Bar graph of the 10 least frequent words.

conciseness and informativeness. By avoiding excessive specificity, captions can effectively capture the essence of an image while remaining concise. In figure 4 we can see that the histogram exhibits a symmetric distribution with a prominent peak centered around a caption length of 10 words.

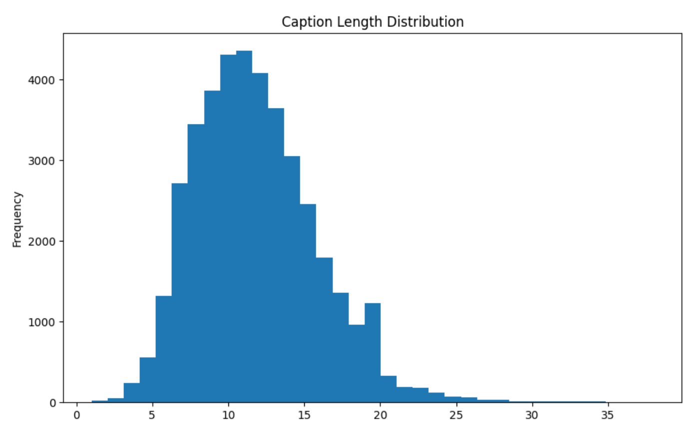

Fig. 4. Histogram of the caption lengths in the flickr 8k dataset.

Part-of-speech tagging.

the Natural Language Toolkit (NLTK), a widely-used Python library, offers a range of functionalities for text processing, including POS tagging. By leveraging NLTK's POS tagging capabilities, we can assign grammatical categories, such as nouns, verbs, and adjectives, to each word in each caption. This process is enables us to comprehend the syntactic structure of the caption and use it to generate more accurate and contextually appropriate descriptions for associated images. In Table I we can see the POS tags with their respective meaning and the frequency of these tags used in the entire dataset in Figure 5.

TABLE I
NLTK POS TAGS

Tag	Meaning
NN	noun, singular (cat, tree)
VBZ	verb, present tense with 3rd person singular (bases)
VBG	verb gerund (judging)
NNS	noun plural (desks)
JJ	adjective (large)
VBP	verb, present tense not 3rd person singular (wrap)
VBN	verb past participle (reunified)
NNP	proper noun, singular (sarah)
VB	verb (ask)
VBD	verb past tense (pleaded)
JJR	adjective, comparative (larger)
JJS	adjective, superlative (largest)
NNPS	proper noun, plural (indians or americans)

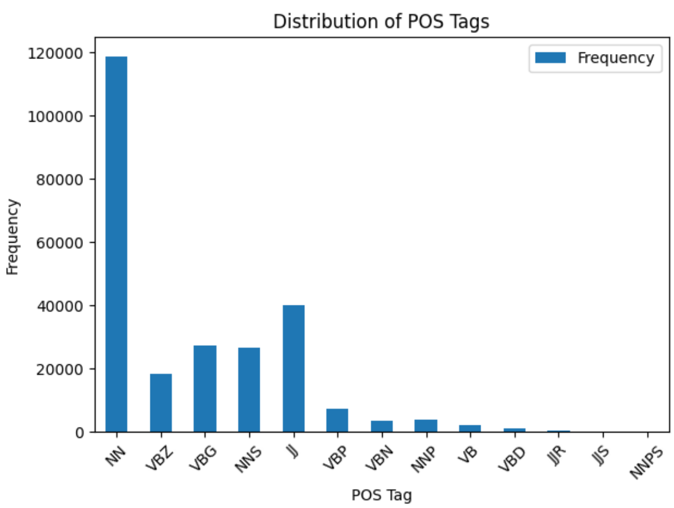

Fig. 5. Bar graph of the POS tag distributions.

B. Model Selection

Model variations. We had the choice between a multitude of different pre-trained models for the sole purpose of feature extraction from our set of training images. However, we developed a shortlist based on the following criteria:

1) **Architecture:** The model's architecture should be well-suited for image feature extraction. We looked for models that have demonstrated strong performance in visual recognition tasks.
2) **Pretrained:** It was important for the model to have been pretrained on large-scale datasets such as ImageNet or COCO. Pretraining allows the model to learn generic visual representations, which can be fine-tuned for our specific image captioning task.
3) **Transfer Learning:** We focused on models that support transfer learning. Transfer learning enables us to leverage the knowledge gained from pretraining on large datasets and apply it to our own task, improving efficiency and performance.
4) **Availability:** We considered models that are readily available and accessible through popular deep learning frameworks like TensorFlow or PyTorch. This ensured ease of implementation and compatibility with our existing infrastructure.

However, availability emerged as a significant constraint during our model selection process. Implementing a state-of-the-art model from an obscure technical paper was not feasible within our time frame. Consequently, we narrowed down our choices to well-known and widely used convolutional neural networks (CNNs) as encoders for image captioning, including VGG, ResNet, Xception, Inception, and DenseNet.Specifically, we tested VGG16, ResNet152, ResNet50, and Xception, evaluating the quality of the generated captions. Our results aligned with the findings in the 2021 paper by Alam et al., where ResNet outperformed the other models. Considering both our experimental results and the literature, we decided to adopt ResNet50 as our feature extraction model. While ResNet50 is shallower than ResNet152, we believed it had better potential for generalization in our transfer learning application. We considered its balance between model depth and computational efficiency to be advantageous for our specific needs. By selecting ResNet50 as our feature extraction model, we aimed to leverage its strengths in capturing visual features and facilitating transfer learning to enhance the quality of our image captioning system.

C. Model Training

Once we selected the ResNet50 model as our feature extraction model, we proceeded to train the image captioning system by combining the extracted image features with a language model component based on LSTM (Long Short-Term Memory) networks.

1) Dataset Preparation: We prepared our training dataset by pairing images with corresponding captions. Each image, reshaped to be of size (224, 224, 3) as is optimal for ResNet architectures, was passed through the ResNet50 model to extract a fixed-length feature vector. The captions were preprocessed by first removing non-alphanumeric characters, extra spaces, and single characters, before being tokenized into wordsand representing them numerically. We also decided to limit our over 8200 word vocabulary to only the top 5000 words as it was observed to offer better performance for image captioning; we think this is in part to the smaller total corpus being easier to handle by our encoder-decoder model structure.

2) Feature Engineering: We prepared our training data set by pairing images with corresponding captions. As mentioned before we reduced the word vocabulary to 5000 words to only consider the most important words. We also converted captions to lowercase, removed non-alphabetic characters, removed extra spaces, removed single-character words, and added ¡start¿ and ¡end¿ tokens to mark the beginning and end of each caption. These techniques helped improve the quality and consistency of the caption text by removing irrelevant characters, reducing redundancy, and enhancing the overall readability. The prepossessed captions can then be used as input for our models.

3) Architecture Design: The architecture of our image captioning model consisted of two main components: the encoder and the decoder. The encoder utilized the pretrained ResNet50 model as the feature extractor. We froze the weights of the ResNet50 layers to preserve the learned visual representations and prevent them from being updated during training. The output of the ResNet50 model was fed into a fully connected layer to reduce the dimensionality of the features and capture high-level semantic information. For the decoder, we used an LSTM network to generate the captions based on the image features. The LSTM was designed to learn the sequential dependencies between words and generate captions word by word. The LSTM network was initialized with an embedding layer to convert the numerical representation of words into continuous vectors. We used Layer Normalization in our encoder Layer to help standardize and normalize the dataset. This helps avoid problems such as exploding gradients and also reduces overall training time. We also used batch normalization [3] in the Decoder Layer which helps normalize inputs throughout the batches. This becomes important as we progress through layers in our model and the input needs to be normalized again.

4) Training Process: We initialized the weights of the LSTM and embedding layers randomly and trained the entire image captioning model end-to-end. The objective was to minimize the discrepancy between the predicted captions and the ground truth captions in the training dataset. We used a variant of the cross-entropy loss, called the categorical cross-entropy loss, to compute the training loss. The loss was calculated between the predicted captions and the ground truth captions, considering each word as a separate class. To optimize the model, we employed the backpropagation algorithm along with the Adam optimizer. We iteratively updated the model's weights by backpropagating the gradients of the loss through the network and adjusting the weights in the direction that minimized the loss.

5) Hyperparameter Tuning: During the training process, we carefully tuned the hyperparameters to achieve the best performance of our image captioning system. Some of the key hyperparameters we considered were:

- Learning rate: We experimented with different learning rates to find an optimal balance between convergence speed and stability.
- Batch size: We varied the batch size to control the number of training examples processed in each training iteration; we found that the larger the batch size the faster time to convergence was, and lower our per-epoch loss.
- Number of LSTM units: We adjusted the number of LSTM units to influence the model's capacity to capture sequential dependencies and generate captions effectively.
- Dropout rate: We applied dropout regularization to prevent overfitting. We tuned the dropout rate to strike a balance between model regularization and preserving important information.
- Training epochs: We determined the number of training epochs, considering factors such as convergence and model performance on validation data.

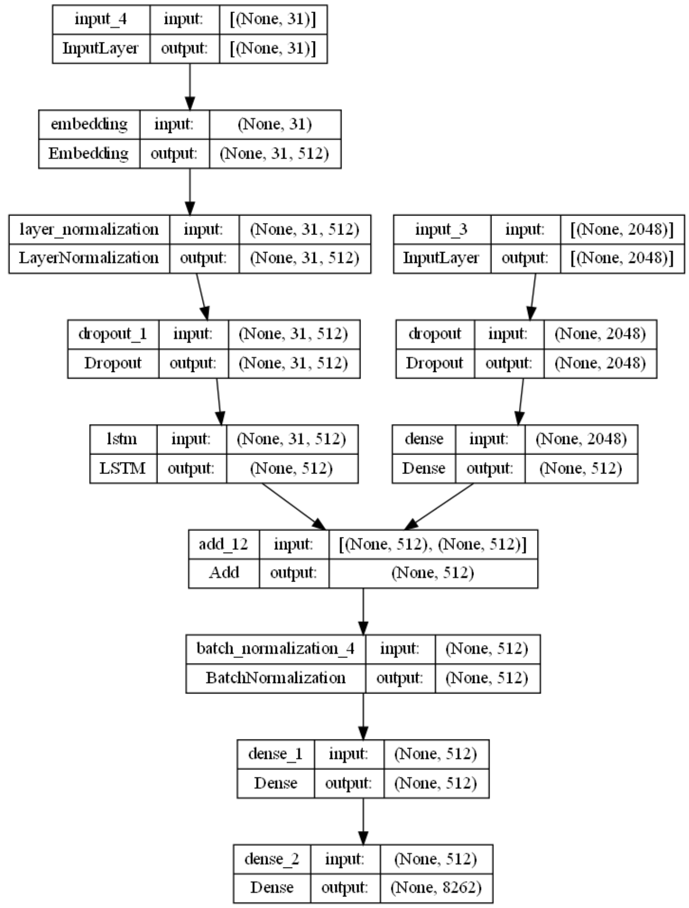

Fig. 6. CNN-RNN model architecture.

6) Model Evaluation: To assess the performance of our trained image captioning model, we utilized a separate validation dataset. During the training process, we periodically evaluated the model on the validation set to monitor its progress and prevent overfitting. For evaluation, we employed a metric commonly used in image captioning tasks: BLEU (Bilingual Evaluation Understudy). This metric provided quantitative measures of the quality and similarity of the generated captions compared to the ground truth captions. By closely monitoring the model's performance on the validation set and iteratively adjusting the hyperparameters, we aimed to achieve the best possible captioning performance and ensure the model's ability to generalize to unseen images.

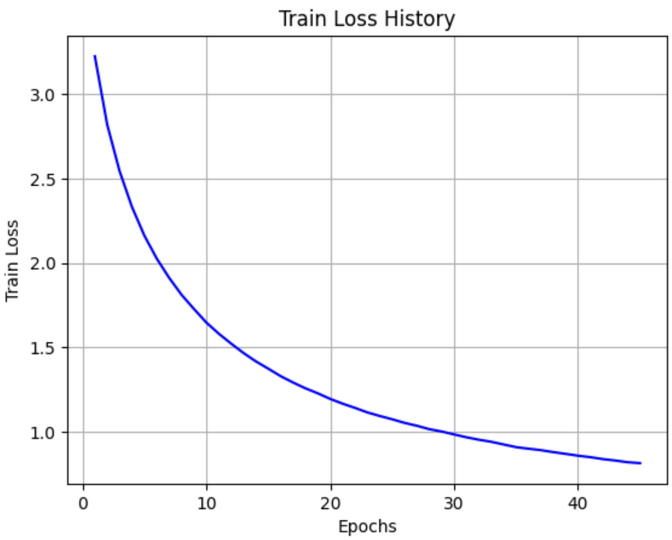

Fig. 7. Training loss history over 30 epochs.

7) Inference: Once the model training was complete, we used the trained image captioning model for inference. Given a new image, we fed it through the ResNet50 encoder to obtain the image features. These features were then inputted to the trained LSTM decoder, which generated a caption word by word. The generated caption was post-processed to improve readability and coherence if necessary. The trained image captioning model was ready for deployment and capable of generating captions for new images, thereby providing a seamless integration of visual information with textual descriptions. By following these steps, we successfully trained an image captioning model using the ResNet50 feature extractor and LSTM-based language model, allowing us to generate captions that capture the content and context of images in a meaningful way.

IV. RESULTS

To predict captions for the images using our trained model we used two decoding algorthims: Greedy Search and Beam Search

- Greedy Search: Greedy search is a simple and efficient decoding algorithm. It works by selecting the word with the highest probability at each step of the decoding process. In caption generation, at each time step, the model predicts the next word based on the current input and selects the word with the highest probability as the output. Greedy search does not consider future possibilities or explore alternative paths, resulting in a locally optimal choice at each step. However, this can lead to suboptimal overall sequences, as the model may get stuck in a suboptimal path early on.
- Beam Search: Beam search is an enhanced decoding algorithm that explores a predefined number of candidate sequences simultaneously. It maintains a set of the top-k most likely partial sequences and expands each of them by considering all possible next words. The expanded sequences are then ranked based on their probabilities, and the top-k sequences are kept for further expansion. Beam search allows for considering multiple possibilities, increasing the chances of finding a globally better sequence. The beam width parameter determines the number of candidate sequences to consider.

We found greedy search to be faster than beam search but beam search provided a much more descriptive. Here are a few of the predictions we made

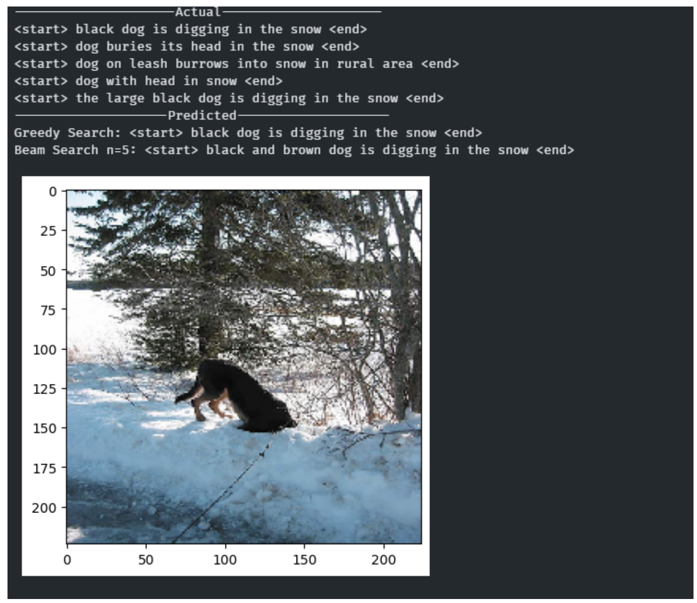

Fig. 8. First Random Prediction.

REFERENCES

[1] A. Jain, "Flickr 8K dataset," Kaggle, https://www.kaggle.com/datasets/adityajn105/flickr8k (accessed Jun. 5, 2023).

[2] Alam et al., "Comparison of Different CNN Model used as Encoders for Image Captioning", ResearchGate, Comparison of Different CNN Model used as Encoders for Image Captioning (accessed Jun. 26, 2023).

[3] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," arXiv.org, https://doi.org/10.48550/arXiv.1502.03167 (accessed Jun. 25, 2023).