Image Captioning

Group members: Yutao Chen, Xiaodan Lu, Ziyue Li, Yunhong Yang

Introduction

Image captioning [1], positioned at the crossroads of computer vision and natural language processing (NLP) in the deep learning [2] field, has emerged as a prominent area of study within artificial intelligence. This technology, aimed at generating descriptive text for images, has considerable implications for enhancing accessibility and user interaction in the digital realm.

Background and Significance

Image captioning, an intricate endeavor bridging computer vision and natural language processing (NLP), has undergone remarkable progress recently, propelled by the overarching development of artificial intelligence. This technology is crucial in converting visual information into descriptive text, yielding substantial benefits in numerous fields.

In today's digital era, characterized by the massive production and consumption of digital content, image captioning is key in content management. It automates the labeling and organization of extensive image libraries, enabling more efficient retrieval and categorization. This is particularly important for digital archives, social media platforms, and online databases, where effective visual data management is paramount.

Beyond this, image captioning's impact stretches into content creation and the media industry. It provides a means for the automatic generation of detailed narratives for images, applicable in diverse contexts such as journalistic reporting, educational materials, and digital assistant applications. As this technology advances, its range of possible uses continues to grow, marking it as an increasingly significant tool in the digital landscape.

Challenges in Image Captioning

While the potential applications of image captioning are vast, the task itself presents numerous challenges that make it a complex problem to solve. At its core, image captioning requires a deep integration of computer vision and NLP to not only understand the contents of an image but also to describe them in a natural and contextually appropriate way.

One of the primary challenges is the accurate recognition and interpretation of the visual elements within an image. This task extends to pinpointing objects, discerning their attributes, and perceiving the interactions and relationships among them. However, recognizing these elements is just the beginning. The model must also interpret them within the larger context of the scene, capturing subtleties such as actions, emotions, and the general narrative, which adds layers of complexity to the task.

Another significant challenge is the generation of coherent and contextually relevant captions. This aspect goes beyond simple object recognition and enters the realm of language generation, where the model must construct sentences that are not only grammatically correct but also contextually fitting with the image. It requires a sophisticated understanding of language and the ability to generate narratives that accurately reflect the visual content.

Additionally, the wide variety and diversity of images contribute to this complexity. If a model is trained on a dataset that is either too narrow or biased, its performance may significantly decline when faced with images that fall outside the range of its training data.

Dataset

About the data

The data used for this project is from the COCO dataset on Kaggle [7]. Developed by Microsoft Research in collaboration with several organizations, COCO is a comprehensive dataset encapsulating a variety of real-world images. It can be used for a comprehensive collection of object detection, segmentation, keypoint detection, and captioning tasks. This extensive dataset contains more than 200,000 images. Its rich content, covering a wide range of scenes, including indoor and outdoor scenes, various objects, and complex interactions, along with captions, makes it a good foundational resource for training and evaluating image captioning models. This project, therefore, builds on this dataset for model training and evaluation analysis.

Data Preprocessing

First, the dataset is preprocessed. Data reprocessing is a key stage in data preparation for machine learning tasks, especially in the context of image captioning. Among the key steps taken in dataset reprocessing include data segmentation, image resizing, data cleaning, and tokenization with padding. The text data was first cleaned with several preprocessing steps, including converting the text to lowercase, removing redundant spaces, eliminating punctuation and special characters, and adding start and end markers. These measures contribute to the overall cleanliness and uniformity of the text data, setting the stage for effective model training.

To build robust models, the dataset was divided into three subsets: the training set made up 80% of the data, the validation set made up 10% of the data, and the testing set also made up 10% of the data. This strategic division into sizes 7228, 903, and 904 ensures an effective balance between model training, validation, and the evaluation of unseen data.

Consistency in input size is crucial for neural networks. During the reprocessing stage, all images were resized to the specified size. This step not only facilitates consistency but also simplifies the training process and ensures that the neural network can seamlessly process inputs of the same dimension.

Next, the text data underwent tokenization, the process of mapping words to integers. This numerical representation allows machine learning models to process textual information. Padding was applied to ensure consistent sequence lengths and to address variations in the length of different subtitles. This ensures that the model can process inputs of uniform dimensions.

In summary, the data reprocessing steps taken are important for the overall quality and applicability of the image captioning task dataset. Together, these enhance the compatibility of the dataset with the neural network architecture. This refined dataset can be used for effective model training and evaluation, laying the foundation for the development of a robust and accurate image captioning system.

Model

Encoder

In this project, the pre-trained model Inception-v3 convolutional neural network (CNN) is used as the feature extractor [3]. This established model is renowned for its ability to extract rich hierarchical features from images. The Inception-v3 model is loaded with pre-trained weights, and its layers are frozen to prevent further training. The input shape for this model is defined as 299*299*3, and the output shape of Inception-v3 is 8*8*2048. The spatial dimensions of 8*8 corresponds to the feature maps, while 2048 signifies the number of filters within each map. Subsequently, the output of the InceptionV3 model is further processed through a dense layer with rectified linear unit (ReLU) activation, resulting in a reduced-dimensional representation referred to as the encoding. To achieve a more compact representation, the encoded output is flattened for subsequent integration into the neural network architecture. This flattened encoding serves as a fundamental input for generating image captions.

Embedding

The embedding layer utilizes an embedding dimension of 256 and incorporates dropout with a rate of 0.15 to prevent overfitting. The embedding layer is applied to the teacher forcing input. The teacher forcing input, caption sequence, is a useful method in training recurrent neural network models that use the ground truth from a prior time step as input. After the teacher force input is embedded, the embedded teacher forcing input and the flattened encoding are concatenated. This operation fuses the spatial information from the encoded image with the semantic information derived from the embedded captions, creating a comprehensive input for subsequent layers of the model.

Decoder

The decoder employs an LSTM model to decode the fixed-length vector, which is derived from the concatenation of both the encoder and embedding layers. Following this concatenation, the resulting vector is fed into the LSTM model to generate a sequence. The decoder, itself an LSTM, has its initial hidden states set by the input layer with 768 units in the decoder layer.

Using these initial states, the decoder initiates the generation of the output sequence. The LSTM model is connected to the concatenate layer and two Input layers, which represent the initial states. At each time step, the LSTM produces an output vector. Finally, all of these vectors are input into a Dense Layer [4]. The output size of the LSTM is a 3D output of the form [(None, 16, 768), (None, 768), (None, 768)].

Since the LSTM decoder is essentially a language model conditioned on the initial states, it is necessary to use a Dense layer to produce the final output [5]. After obtaining an output from the LSTM model and feeding it into the Dense Layer, which functions as the final layer to generate the sequence, the critical aspect in the Dense layer is the number of units, determining the shape of the output vector [4]. The output of the Dense Layer represents the probability of each word. Finally, these probabilities are employed to obtain the index of the tokenizer, leading to the generation of a sentence that serves as the caption.

Loss Function

The loss function plays a significant role in the backpropagation training of a model, and choosing a better loss function contributes to improved model training. Cross-entropy is commonly used in classification tasks, and in this project, it serves as the appropriate loss function for model training.

During the data preprocessing stage, adding padding values to achieve a uniform size enhances the model's performance. Therefore, when using the loss function, it is essential to disregard the loss contribution from padded values. Additionally, for cross-entropy, it is crucial to employ one-hot encoding for the observed values, which involves assigning a unique one-hot vector to each class. Finally, the predicted values are used to compute the cross-entropy loss between the predicted and true values.

Evaluation

In this project, three pivotal evaluation metrics for language processing models are outlined: BLEU, METEOR, and ROUGE scores. ROUGE is used to gauge the quality of text summarization by measuring the overlap of N-grams and Longest Common Subsequence between system-generated and reference summaries. BLEU is a standard metric for machine translation, assessing the precision of N-grams between candidate translations and reference texts. METEOR, also for machine translation, calculates the harmonic mean of unigram precision and recall, factoring in a penalty for discrepancies in length between the model's output and the reference translation [6]. These metrics are essential tools for quantifying and improving the performance of language models by comparing their output to that of humans.

Results

Training and Validation Loss (Figure 1): The left graph shows the training loss (in blue) and validation loss (in orange). Loss is a measure of how well the model is performing; a lower loss indicates better performance. As the number of epochs increases, the training loss decreases steadily, which suggests the model is learning from the training data. The validation loss also decreases initially but then plateaus, indicating that the model is not improving on the validation data after a certain point. This could be a sign of overfitting, where the model learns the training data too well and fails to generalize to new, unseen data.

Training and Validation Accuracy (Figure 2): The right graph displays the training accuracy (in blue) and validation accuracy (in orange). Accuracy is the proportion of true results (both true positives and true negatives) among the total number of cases examined. The training accuracy increases over time, suggesting the model is getting better at correctly identifying or classifying the training data. However, the validation accuracy increases initially and then levels off, which could indicate that the model's ability to generalize to new data is not improving significantly after a certain number of epochs.

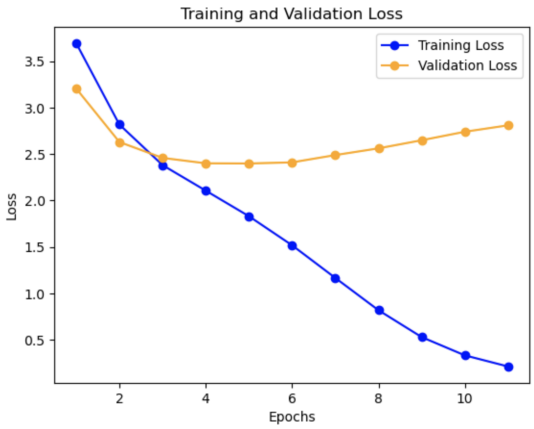

Figure 1.

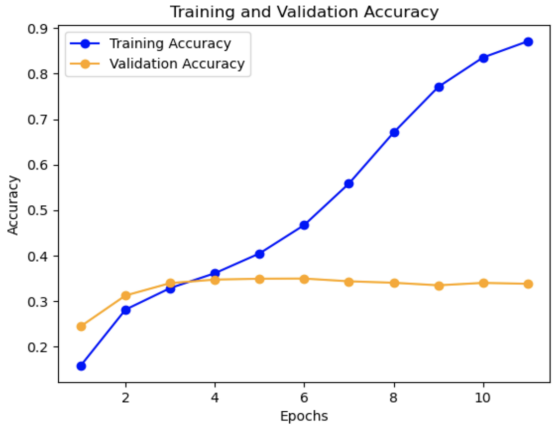

Figure 2.

A bar graph (Figure 3) illustrates the average scores: a BLEU score of 0.1028 reflects lower exact match precision, while METEOR and ROUGE scores, at 0.3370 and 0.3086 respectively, on the test set, show a stronger grasp of meaning and structure in the model's output. These metrics suggest that the model is better at understanding and reproducing the intent and context of the text rather than producing word-for-word translations.

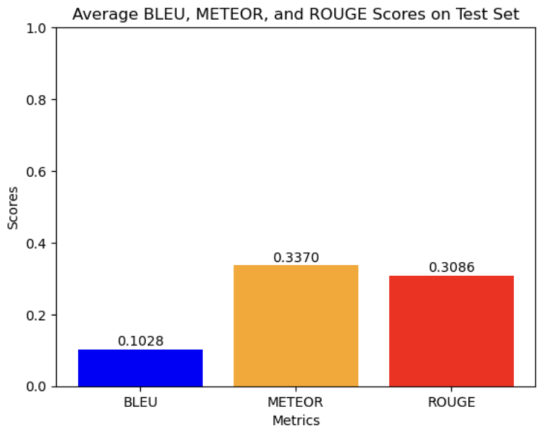

Figure 3. Bar graph with average BLEU, METEOR, and ROUGE score

A few examples are presented in **Figure 4**, utilizing the model for predictions on the test set, elucidating the distinctions between the original captions and their corresponding predicted counterparts. This effectively showcases the model's capacity to generate captions across diverse visual contexts. While the generated text may exhibit slight deviations from the original captions, it generally demonstrates an apt understanding of the image's content, providing comprehensible and reasonable descriptive text. However, there is room for improvement in refining the model to deliver even more accurate textual outputs. This comparative analysis thus unveils both the model's strengths and areas for enhancement, contributing to a more profound comprehension of its caption generation capabilities.

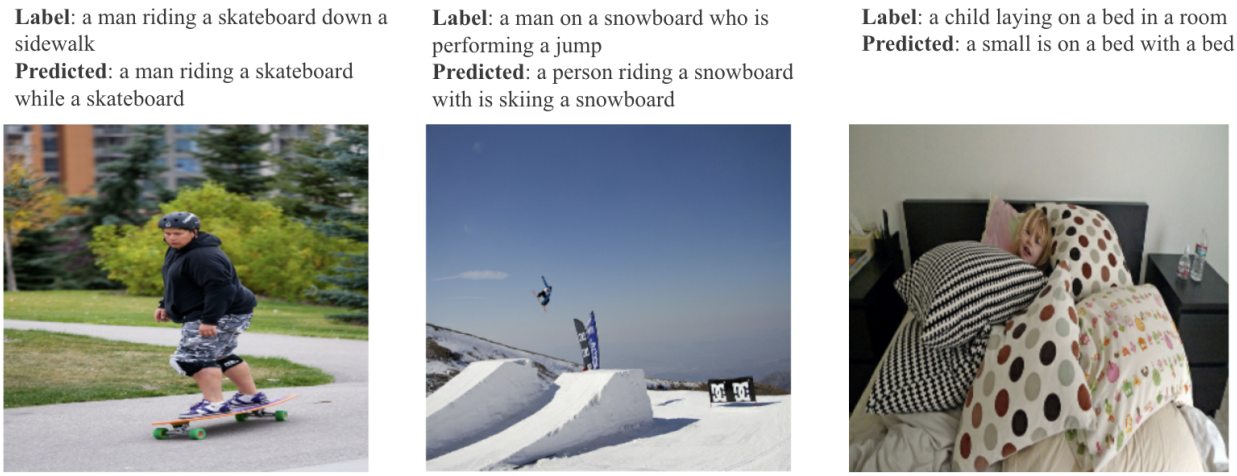

Figure 4. Examples comparing the original captions to the predicted captions

Conclusion

This project underscores the importance of dataset selection in model training. The diversity and richness of the COCO dataset played a pivotal role in the model's learning process.

Future OutlookLooking forward, there are several avenues for further research and development:

- Exploring More Diverse Datasets: To enhance the model's robustness and applicability, it would be valuable to train and evaluate it on other datasets, especially those that differ significantly from the COCO dataset in terms of image and caption styles.
- Refining the Model’s Architecture: Based on our model we can adjust the number of layers or the size of the models, which could lead to further improvements in performance.

In conclusion, this project represents a significant step in the field of image captioning, demonstrating the potential of combining advanced vision and language models. The insights gained and the challenges identified.

References

[1] Q. You, H. Jin, Z. Wang, C. Fang, and J. Luo, "Image Captioning with Semantic Attention," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 4651-4659, doi: 10.1109/CVPR.2016.503.

[2] LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* 521, 436–444 (2015). https://doi.org/10.1038/nature14539

[3] Szegedy, Christian, et al. "Rethinking the inception architecture for computer vision." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

[4] BM, Nechu. “How to Build an Encoder Decoder Translation Model Using LSTM with Python and Keras.” Medium, Towards Data Science, 21 Oct. 2020, towardsdatascience.com/how-to-build-an-encoder-decoder-translation-model-using-lstm-with-python-and-keras-a31e9d864b9b.

[5] "Dhote, P. (2020, August 20). Seq2Seq-encoder-Decoder-LSTM-model. Medium. https://pradeep-dhote9.medium.com/seq2seq-encoder-decoder-lstm-model-1a1c9a43bbac

[6] Mansuy, R. (2023, November 27). Evaluating NLP Models: A comprehensive guide to ROUGE, BLEU, METEOR, and BERTScore metrics. Medium. https://ai.plainenglish.io/evaluating-the-performance-of-natural-language-processing-nlp-models-can-be-challenging-ce6f62c07c35

[7] Awsaf. “Coco 2017 Dataset.” *Kaggle*, 4 Sept. 2020, www.kaggle.com/datasets/awsaf49/coco-2017-dataset.